Instrumentation of a Skeleton Sled: Novel Tactile Steering Force Sensors

By

Camilo Andrés Rachello
4695313

in partial fulfillment of the requirements for the degree of

Master of Science
in Mechanical Engineering Biomechanical design
track Sports Engineering Specialization

at the Delft University of Technology,
to be defended publicly on 18th of December, 2019

Supervisor:	Prof. Dr. Ir. Arend Schwab	
Thesis committee:	Prof. Dr. Ir. Dick Plettenburg,	TU Delft
	Prof. Dr. Ir. Volkert van der Wijk,	TU Delft

An electronic version of this thesis is available at http://repository.tudelft.nl/.

TU Delft

Contents

Abstract 4

Acknowledgments 5

1. Introduction 6

1.1. Skeleton 6

1.2. Problem description 7

1.3. Research question 7

1.4. Thesis outline 8

2. Methods 8

2.1. Steering force sensors 8

2.1.1. Sensing method evaluation 8

2.1.2. First Prototype 9

2.1.3. Full-size sensor calibration test 12

2.1.4. Full-size sensor calibration results 14

2.1.5. Final design 19

2.2. Data collection and processing 21

2.3. Placement on the sled 22

3. Results 24

3.1. Tests 24

3.2. Tests Results 26

3.2.1. Test 1 – Fixed Weights 27

3.2.2. Test 2 – Constant Force 28

3.3. Steering Force Models 31

3.3.1. Left Shoulder Model 31

3.3.2. Right Shoulder Model 32

3.3.3. Left Knee Model 32

3.3.4. Right Knee Model 33

3.3.5. Validation test – Free steering Forces 34

3.4. Presentation to the athlete 37

4. Analysis and Discussion 37

4.1. Analysis of the Steering Force Models 37

4.2. Accelerometers Integration with the Steering force sensors 39

5. Conclusion 40

5.1 Future Work 41

Appendix A – Fixed Weights 42

Appendix B – Constant Force 43

Appendix C – Free Steering 47

Appendix D – Arduino Code 52

Appendix E – GUI Interface 55

References 58

Abstract

The purpose of this project is to design and develop a set of force sensors to measure steering forces applied by an athlete down the ice track. Currently, there is not enough information about instrumentation in skeleton, and to maintain competitive advantages, most of the research remains private and unpublished. Athletes use their shoulders and knees to steer down the track. For this reason, four handmade piezoresistive tactile force sensors were built to measure the force applied by each joint. Each sensor has its own model to convert bits recorded into force. Results showed a difference between applied and calculated forces by each model. However, calculated results followed similar trends compared to the real values of the applied force. In addition, a graphical user interface was created to present the results to the athlete in a simple and easy way to read and understand. It is planned to use a shimmer (an inertial measurement unit) to collect information about accelerations developed on each run. Coupling both measurement systems have to be done during the processing stage. Further work has to be done regarding electronics size and testing the systems in a real run down the track.

Acknowledgments

I would like to give a special thank you to my supervisor, Dr. Ir. Arend Schwab. Without all his feedback, patience and good directions, I would not be able to finish this project the way it was finished. When we accepted this project, it was a bold move because Skeleton was a new sport for all members of the project, but we learned and we made this project possible. Also, together with Sanjit Shankar, we battled through this project since the beginning, trying to understand all the aspects of a sport very foreign to us. It was a great journey, I learned so many new things and we had the opportunity to work next to an Olympian athlete, which for a sports engineer is a dream come true.

Moreover, I would like to thank my family, especially my parents and my aunt, for all the support they gave me during this long year of hard work. I understand all the stress they had to go through when things were not working out here very well, or when I had to restart the process every time I failed trying a new idea. However, they remained positive and they were always there for me, cheering me up, giving me all their love and good energy and trying to calm me down when things were not going the way I wanted.

Finally, I want to thank my girlfriend. She had to deal with me and my stress during the last months of work. She was always there, ready to listen to the things that were going wrong. With a lot of patience and good advice, she helped me get over those little obstacles and kept my hopes high that I was going to finish strong. She is the best!

1. Introduction

1.1. *Skeleton*

Skeleton is one of three Olympic winter sliding sports (along with bobsled and luge). Skeleton consists of sliding down a man-made ice track. Sliders lie face down on top of a metallic sled and start descending with their heads first. The length of the ice track is between 1.2 and 1.8 km and it has different corners, curves, and bends. Elite athletes can reach speeds between 115 and 150 kph and up to 5g of force in the corners [1]. Usually, sliders have several runs in each race, the winner is the one who has the fastest combined time over different runs [2]. A single run consists of three stages: The push start consists of the 15-meter sprint which the sliders must accelerate to reach the maximum speed possible, then they have to dive face-down and headfirst into the sled to get a race position and finally, start their descent to the finish line [3].

The difference between the three Olympic sliding sports is that skeleton is the only one that does not involve more than one slider. Also, it was the latest of the three to be included permanently in the Winter Olympic Games in 2002.

The sled consists of three major parts: 1) a frame, 2) a belly pan and 3) a pair of runners (Figure 2.1). The frame is a rectangular steel construction approximately shoulder width and shoulder to knee length [1]. The belly pan is located at the bottom of the sled and consists of a fiberglass or carbon fiber shell that covers the steel frame. Finally, the runners are circular, made out of steel, fixed on the left and right side underneath the sled, and they have two grooves that create a spine in half of its full length in order to dig into the ice [4].

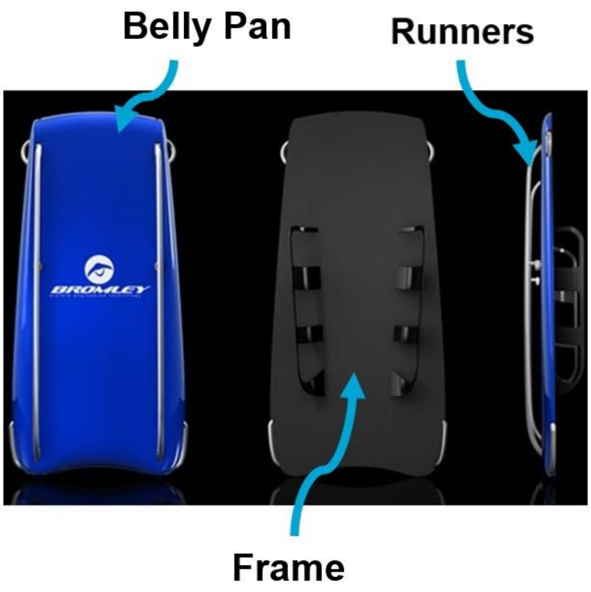


Figure 1.1 Parts of the sled [5].

The sled is regulated by the IBSF (International Bobsleigh and Skeleton Federation) and the weight and dimensions are described as follows in Table 1.1[6]:

Table 1.1. Sled regulations

Weight (kg)				**Dimensions (mm)**
Max combined Weight		**Max Sled Weight**		Length: 800 – 1200
Men	Women	Men	Women	Height: 80 - 200
115	92	43	35	Distance between runners: 340 - 380

Talking about the steering, there are three ways to steer the sled through the descend. The first one is tilting the helmet to create lateral drag force making slight changes of direction. The second one is dragging the toes into the ice. This way of steering is to make abrupt changes of direction and it is used in emergency cases when the slider needs to fix his course while descending. The last one is shifting the weight distribution on the sled in order to make one of the spines dig more into the ice than the other allowing the slider to change the direction of the sled [4]. This steering type is achieved by putting pressure with only shoulders, only knees or a combination of both. Usually, the sliders describe the latter as an ‘X’ because they steer using opposite sided (crossed) joints. This means that when they use the left shoulder and right knee or vice versa.

1.2. Problem description

Since Skeleton is a recently added sport to The Winter Olympic Games, the interest of researching and improving athlete’s performance is growing. However, at the moment there is not a lot of information published and what it is available, is more focused on enhancing push-start performance, ice-runner interaction and aerodynamics of the descend.

Research in instrumentation in skeleton has not been well developed and it is a big field that could help understanding how sliders perform during the descent, especially in a sport in which small-time margins could have a big impact on the final results. Having a better understanding of the descent could bring benefits to the athlete such as better preparation and specific skill training, address issues that could bring a reduced time during the descent and understand how much force he is applying entering and exiting the corners. There are some restrictions on the sport that limit the use of technology aids. For example, for safety reasons, nothing could be on top of the foam which they lay on. Also, measurement systems can be used only during training runs and not in official competition. For these reasons, a new type of steering force sensors needs to be developed and implemented.

1.3. Research question

This project aims to design and build steering force sensors. These novel sensors along with a triaxial accelerometer will be used to instrument a skeleton sled in order to give Akwasi Frimpong a better understanding of his performance during his descent on the track. Several research questions were formulated to achieve the ultimate goal of the project.

1. What type of tactile sensor could work to measure the steering force applied by the athlete?
2. Where and how the sensors will be placed on the sled?
3. How will the data be collected, processed and presented to the athlete?
4. How to combine the force sensors and the inertial measurement unit?

1.4. Thesis outline

After the topic, problem statement and research questions were explained, this thesis will consist of the following parts: 1) the methodology followed will be described. The sensing method chose, the preliminary designs of the sensors, early tests and calibrations of the prototype, the final design developed and the electronics picked are included in this section. 2) The results of the final design are presented. 3) A discussion of the results is shown. In this section, it is analyzed how well the models developed for each sensor fit compared to reality. Finally, 4) the conclusions and future are listed and explained. In the appendices, the complete results of the missing trials are presented, as well as the code for the steering force sensors and the app that shows the results to the athlete.

2. Methods

2.1. Steering force sensors

In the following section, the whole methodology and design process of the steering force sensors will be explained. It will begin by describing the design ideas for the first prototypes and how they were tested. Then, the process for building the real size sensors that will be installed on the sled will be described. Additionally, all the calibration tests will be explained as well.

2.1.1. Sensing method evaluation

Research on tactile force sensors has been actively conducted and led to three sensing methods: capacitive, piezoelectric, piezoresistive. All three solutions have the same structure, consisting of two metallic electrodes with sensing material in between. The capacitive sensing method is based on the parallel plate capacitor. When a force is applied the sensing material compress and its capacitance changes (in the order of femtofarads) and this change can be related to the load applied. The capacitive sensors have high sensitivity due to the small changes in capacitance, but their electronics are complex in order to read these changes (Figure 2.1b, Table 2.1) [7][8][9].

The piezoelectric method uses a Polyvinylidene fluoride (PVDF) in between the metallic electrodes and this material produces a voltage when a force is applied. This type of sensors only work for dynamic applications, but their main advantages are that the system could be self-powered and its high sensitivity (Figure 2.1c, Table 2.1). Finally, the piezoresistive method measures the change in resistance of the sensing material when a load is applied. This solution, compared to the capacitive and piezoelectric solutions, has easier conditioning electronics and readout circuit and simpler design (Figure 2.1a, Table 2.1) [7][8][9]. Also, it is low cost due to the simple materials used was a factor considered in the selection of this sensing type. Additionally, it has a high spatial resolution and can be used in dynamic and static applications [10].

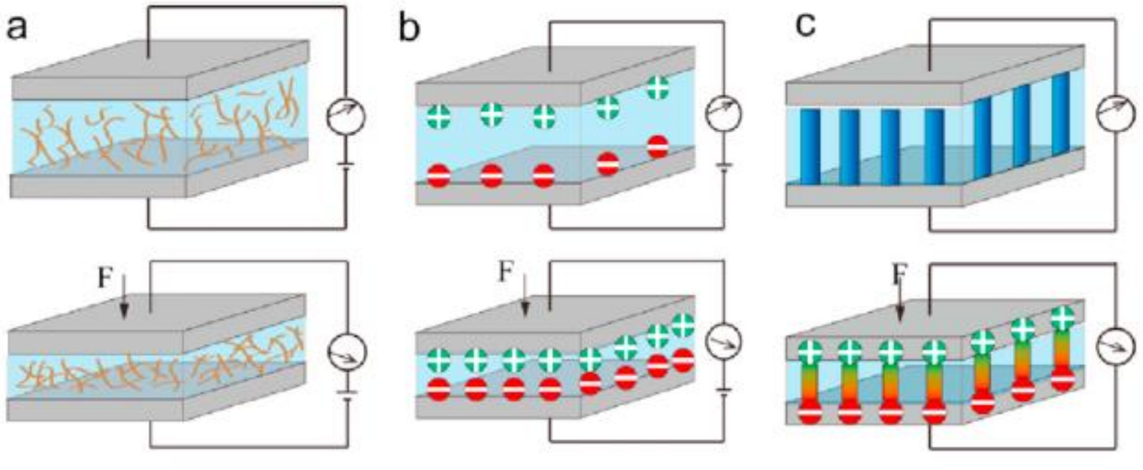


Figure 2.1. Three sensing methods considered. a) Piezoresistive, b) Capacitive and c) Piezoelectric. [10]

Table 2.1. Advantages and disadvantages of the sensing methods considered

Type of Sensor	Advantages	Disadvantages
Piezoresistive	- Simple electronics - High spatial resolution - Low cost - Simple materials	- Hysteresis - Power consumption
Capacitive	- High sensitivity - High spatial resolution - Large dynamic range - No temp dependant	- Complex electronics to measure femto farads - Cross-talk - Hysteresis - Susceptible to noise
Piezoelectric	- High Freq. response - Self-powered - High accuracy - High sensitivity - High Dynamic Range	- Poor spatial resolution - Charge leakages - Only Dynamic

2.1.2. First Prototype

After picking piezoresistive sensors as the desired sensing method, the designing process started. The first step was to make a small prototype to see its behavior after applying forces to it. The general idea for the small prototype was the same for all, but different versions with slight changes were developed. In general terms, it will consist of one piece of piezoresistive material (Velostat) covered with a spacer, in between two metallic electrodes, two pieces of nonconductive material and two metallic buttons to connect the sensor to the microcontroller. The different versions only varied in the type of spacer, the nonconductive fabric and the type of electrode. The same Velostat and metallic buttons were used in all versions and the different designs are described as follows: Two pieces of conductive fabric as electrodes sewed it to a cotton fabric with a thin fiberglass mesh as a spacer

- Two pieces of conductive fabric as electrodes sewed it to a cotton fabric with a nylon mesh as a spacer

- Two pieces of conductive fabric as electrodes sewed it to a cotton fabric with a thin fiberglass mesh as a spacer (Figure 2.2)
- Two pieces of conductive fabric as electrodes sewed it to sponge cloth with a thin fiberglass mesh as a spacer (Figure 2.2)
- Two pieces of conductive fabric as electrodes sewed it to sponge cloth with a nylon mesh as a spacer
- Multiple conductive threads as electrodes sewed it to a cotton fabric with a thin fiberglass mesh as a spacer
- Multiple conductive threads as electrodes sewed it to a cotton fabric with a thin nylon mesh as a spacer
- Multiple conductive threads as electrodes sewed it to sponge cloths with a fiberglass mesh as a spacer (Figure 2.2)
- Multiple conductive threads as electrodes sewed it to sponge cloths with a nylon mesh as a spacer

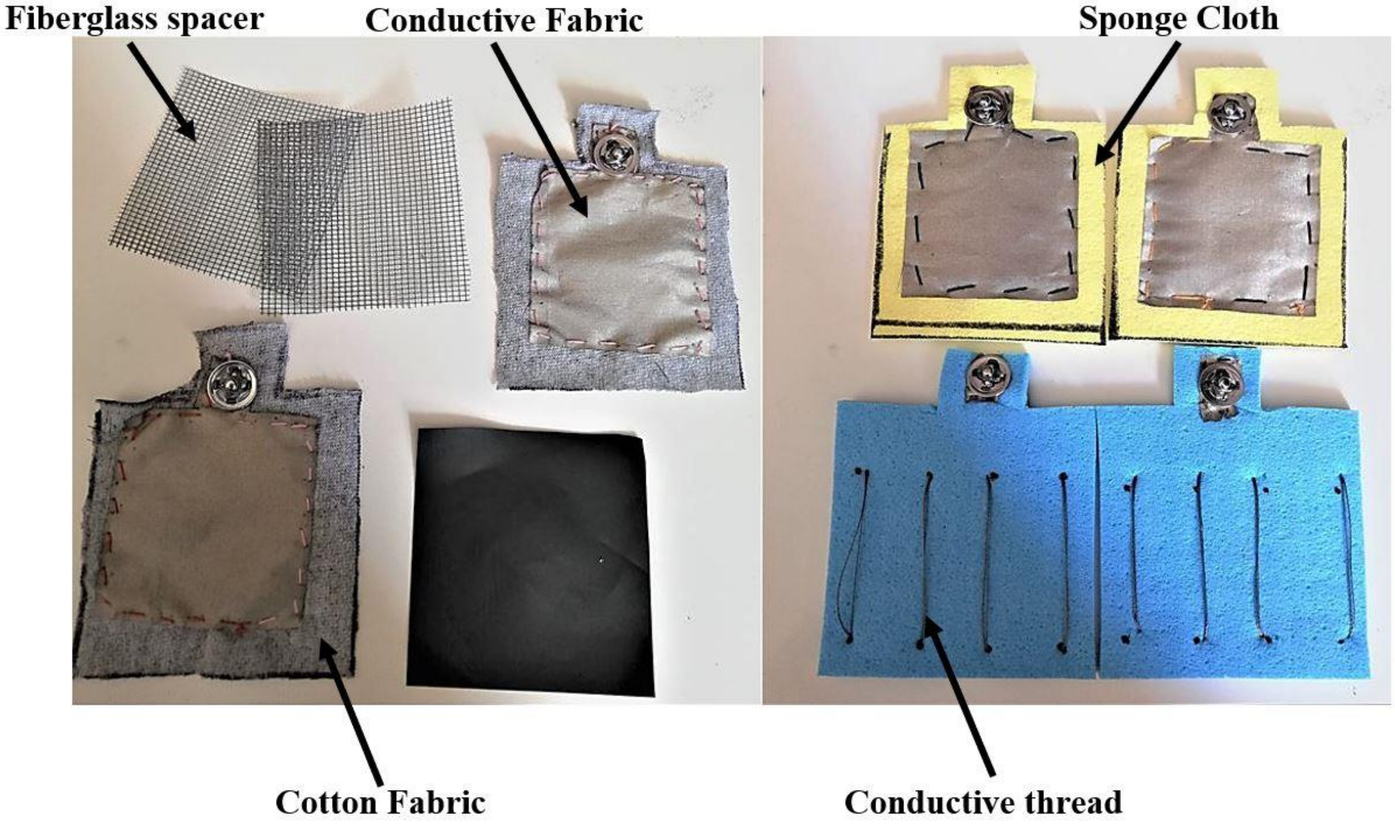


Figure 2.2. Examples of some of the prototypes developed

After building the different versions of the first prototype, the next step was to do a static calibration test to see its response and material behavior. The test consisted of loading the sensor with different weights (from 0 kg to 33 kg) and record its response. The sensor was placed on a flat surface and a flat square metallic base was used to place the oversized weights. An aluminum rod was used as a guide to fit the slots of the weights to keep them in the same position over the different trials. The sensor was connected to an Arduino MEGA 2560 and the results were registered manually. The input to the Arduino is voltage and output are bits that later will be plotted to see the behavior of the sensor (Figure 2.3).

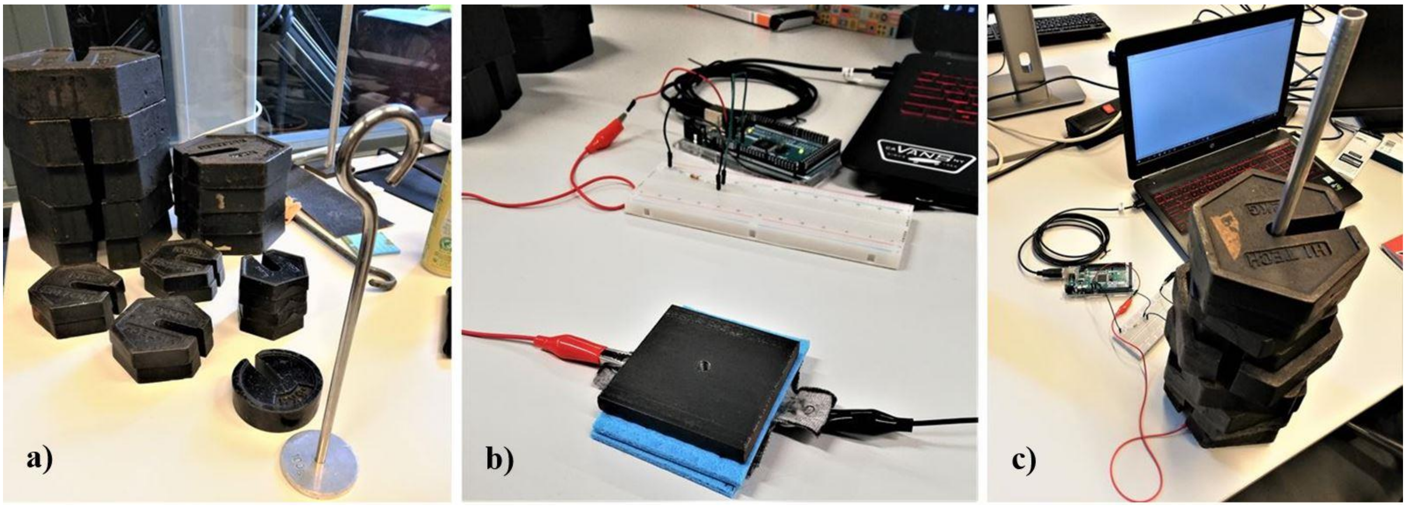


Figure 2.3. Calibration test set up for the prototypes. Figure a) shows the set of weights used for the calibrations. b) Small squared based used to place the weights and c) Set up of the test.

The response of the prototype was a very good first approach of what it was expected from the sensor. Its behavior was linear, which means that the load and the number of bits increased proportionally. However, some versions of the prototype were discarded because of different reasons; some materials such as the fiberglass mesh or the sponge cloth, absorb part of the load and the sensor could not make a proper read, making it less sensitive to the applied loads. Also, the conductive thread did not make enough contact with the sensing material and no readings were recorded. Since the size of the sensor was small and the weights were oversized compared to it, placing the weights was tricky and could affect the readings of the sensor, making the force more specific than distributed.

In the end, it was decided to make a real size version of the prototype with conductive fabric as electrodes sewed to two pieces of microfiber cloth, Velostat with a nylon fabric mesh as a spacer and two metallic buttons sewed with conductive thread that will act as conductive connectors (Figure 2.4). The response to the forces applied to this design was linear, meaning that with an increase in force, more bits will be shown by the sensor (Figure 2.5).

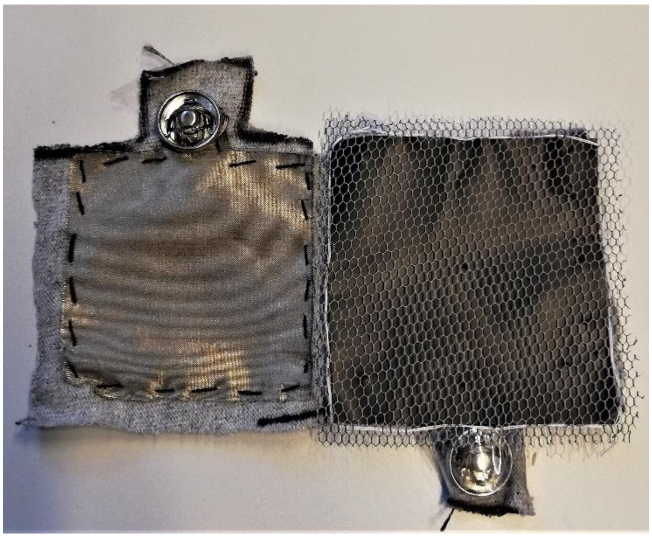


Figure 2.4. Final prototype design selected

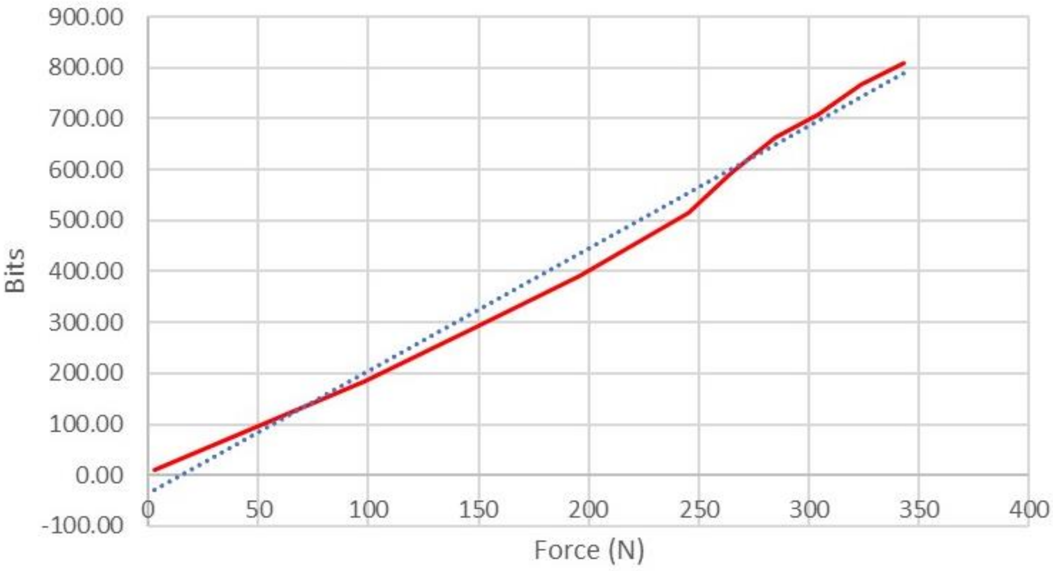


Figure 2.5. Bits vs. Force. Calibration results for the final prototype design selected

2.1.3. Full-size sensor calibration test

For designing the final version of the sensor, a full-size prototype was developed. To know the approximate dimensions for this prototype, the athlete was requested to send pictures of where he places his shoulders and knees and approximate area of contact between his shoulders, knees and the sled (Figure 2.6). This was a good first approach to test the behavior of the real size sensor and what should be the modifications to the design. Each shoulder sensor is 180x180 mm and for the knees, each one is 150x150 mm. In the end, four sensors, one for each joint, will be developed. However, for this first full-size calibration, only one a shoulder size sensor was built and tested (Figure 2.7).

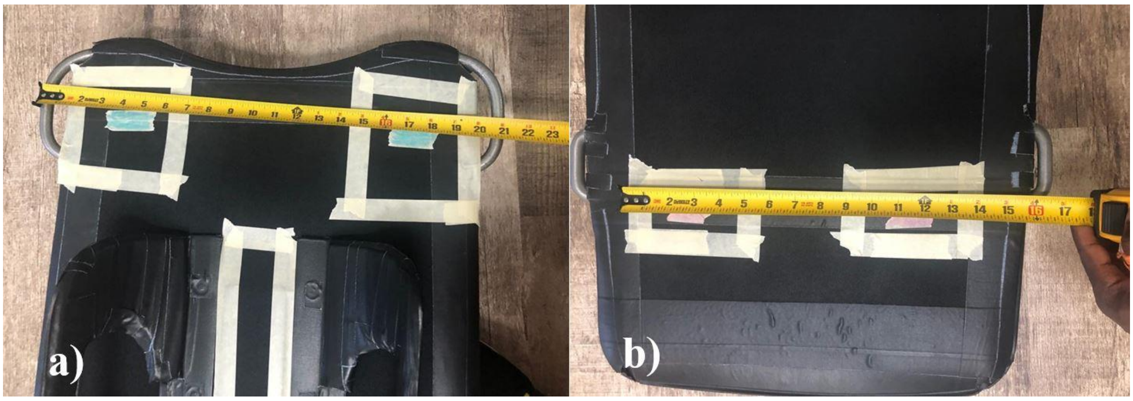


Figure 2.6. The dimension of the sensors. In figure a) and b) it is shown an approximation of the size of the sensors for the shoulders and the knees respectively sent by the athlete. The metric tape is in inches.

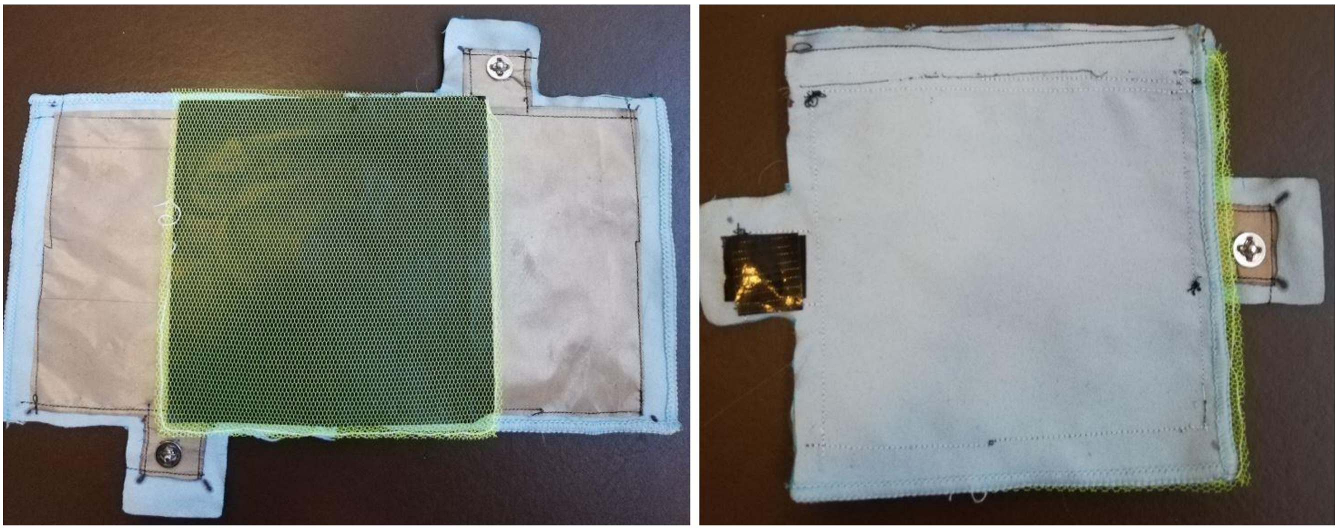


Figure 2.7. Full-size prototype. The left picture shows the sensor's inside with the conductive fabric and the Velostat within the mesh fabric acting as a spacer.

The calibration consisted of applying a compressive force on a tensile bench (Lloyd Instruments LR 5K) using six indenters of different diameters, which are 20, 30, 56, 79, 99 and 115 mm. The maximum force applied was 1250 N and it was calculated as follows:

$$F_{max} = \frac{Body\ Weight}{4} * 5g = \frac{90kg}{4} * 5 * 9.81 = 1104\ N$$

Akwasi's weight is 88 kg so it was rounded to 90 kg and it was divided by four because it is assumed that his body weight will be distributed equally over the four sensors. The 5g corresponds to the maximum g forces that the athlete can experience during the descent in Skeleton. The three smaller diameters (56, 30 and 20 mm) could not support this maximum load, so trial tests were conducted with each one to see which would be the maximum force applied. As a result, 900, 800 and 500 N respectively were the maximum force applied for each diameter.

Finally, the rest of the tensile bench parameters were set. The speed of the machine was 0.025 mm/s and a preload of 25 N was adjusted to shorten the time of each trial. Five trials per diameter were executed and each trial lasted around 3 minutes. Each trial was stopped when the sensor saturated, this means that for a change in force applied, the number of bits remained constant or had a minimal change. The sensor was placed on the bench between two foam mats used to cover the top of the sled and a break of five minutes between trials was given to the foam to recover its original shape after the compression (Figure 2.8).

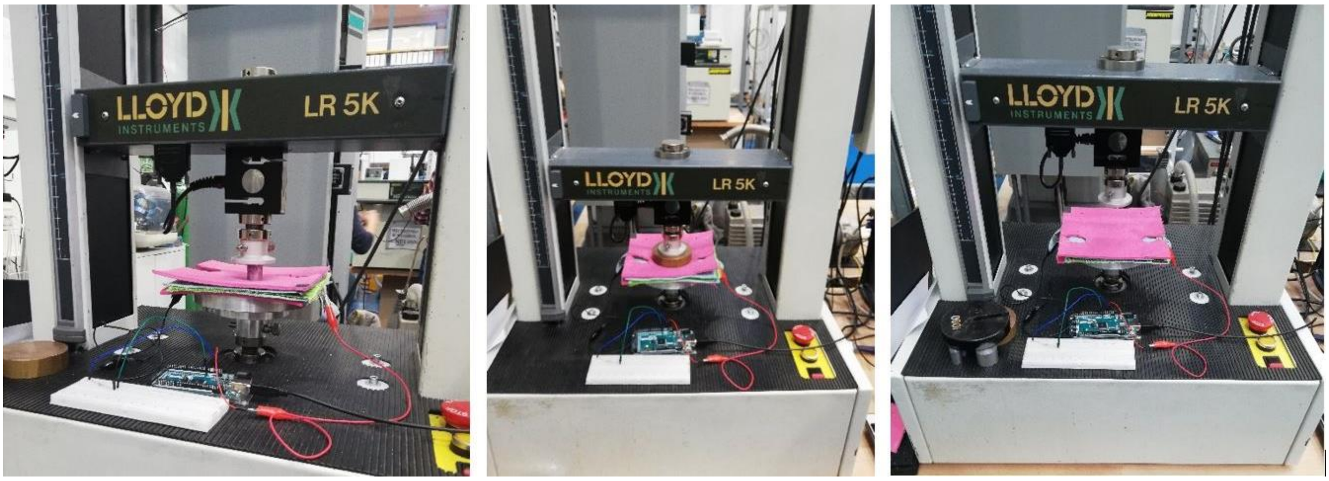


Figure 2.8. Calibration set up for three different diameters.

2.1.4. Full-size sensor calibration results

The data collected was 2 files for each trial: one from the tensile bench with the load applied and its timestamp and another one from the Arduino and the sensor with the bits corresponding each load. The sensor responded differently according to the diameter and the load applied. For the three smallest diameters, the sensor started recognizing force at lower loads, but their range was limited getting saturated quickly. On the contrary, the three largest diameters took a longer time to recognize the force, but their range was bigger and getting saturated close to the maximum force (~ 1000 N).

The first step was identifying at what instant of time the sensor start recognizing the force applied and match that with the file from the tensile bench. After plotting the 5 trials, an average of the trials was calculated for each diameter. Since the size of the data for each trial is different from each other, interpolation and extrapolation were executed to get an equal number of data points for all trials (Figure 2.9 to Figure 2.14). After getting these data points, the plot will be the reference to analyze the behavior of the sensor when different loads were applied to different diameters of contact areas.

All 6 tests had similar behavior (Figure 2.9 to Figure 2.14). It starts with a linear response within the force range for each diameter and when that range is surpassed, the sensor saturates and the change in bits is minimal. It is also clear that the range where the sensor has a linear behavior varies from diameter to diameter. The largest diameter has a longer range and less steep slope and the smallest diameter has a limited range and steeper slope.

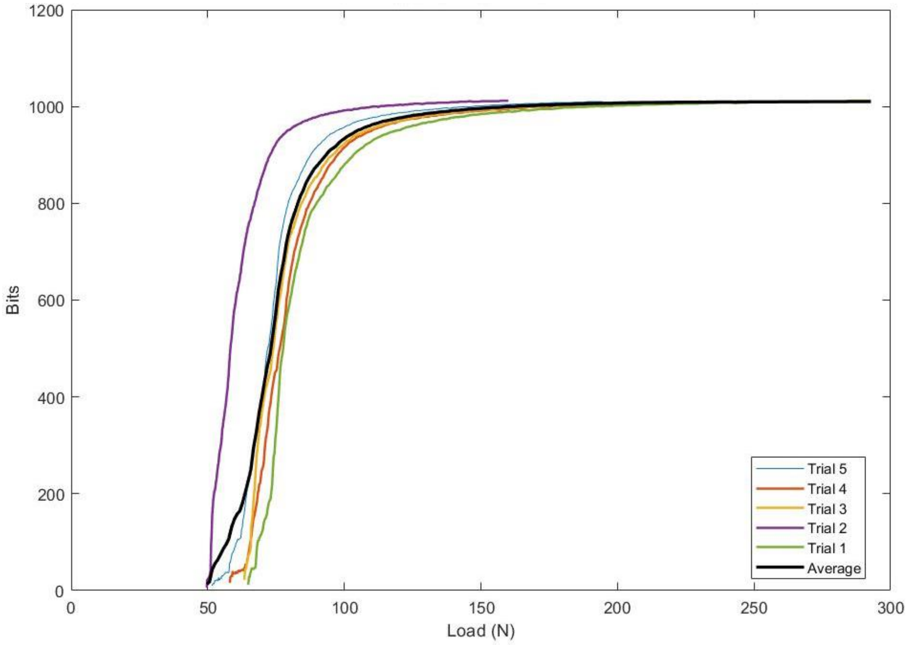


Figure 2.9. Bits vs. Load for 20 mm of diameter. All five trials and their average are plotted.

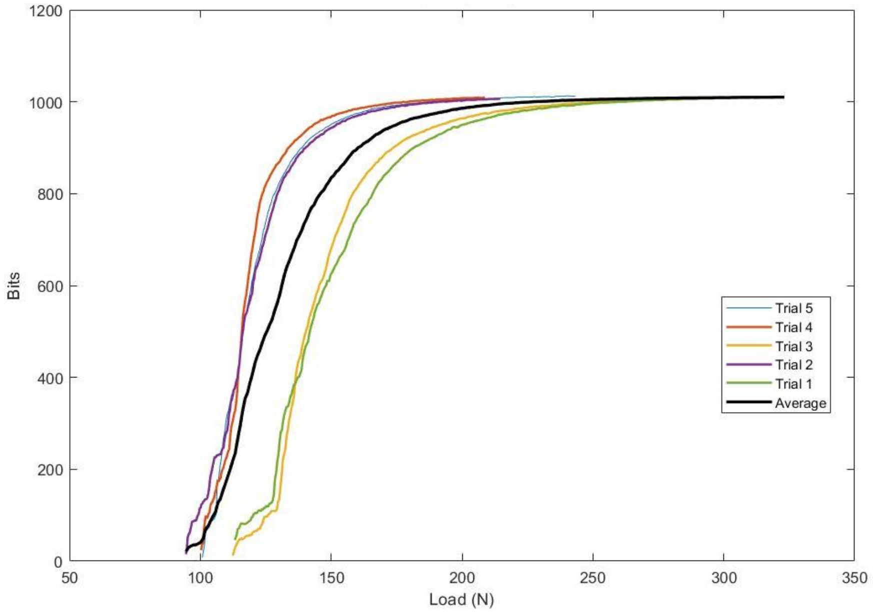


Figure 2.10. Bits vs. Load for 30 mm of diameter. All five trials and their average are plotted.

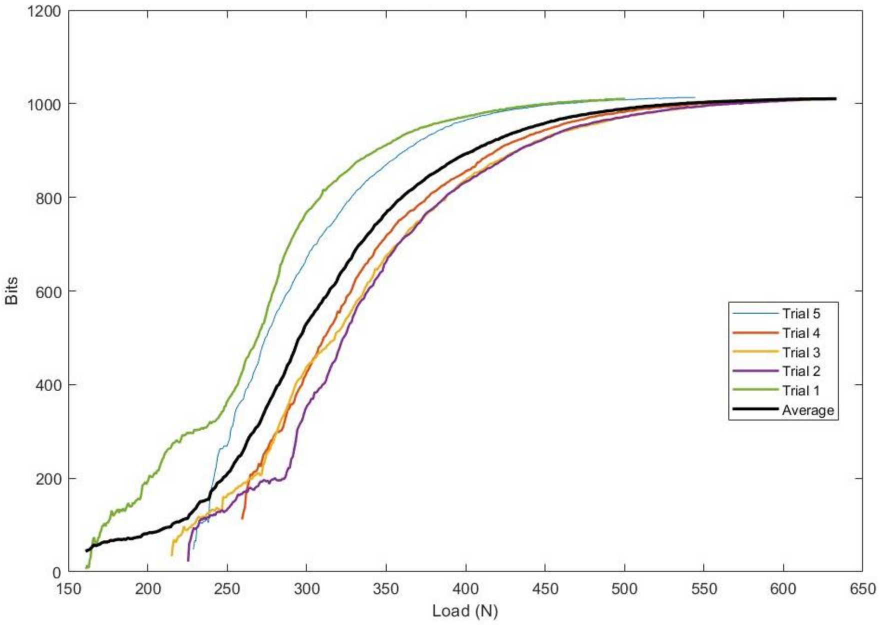


Figure 2.11 Bits vs. Load for 56 mm of diameter. All five trials and their average are plotted.

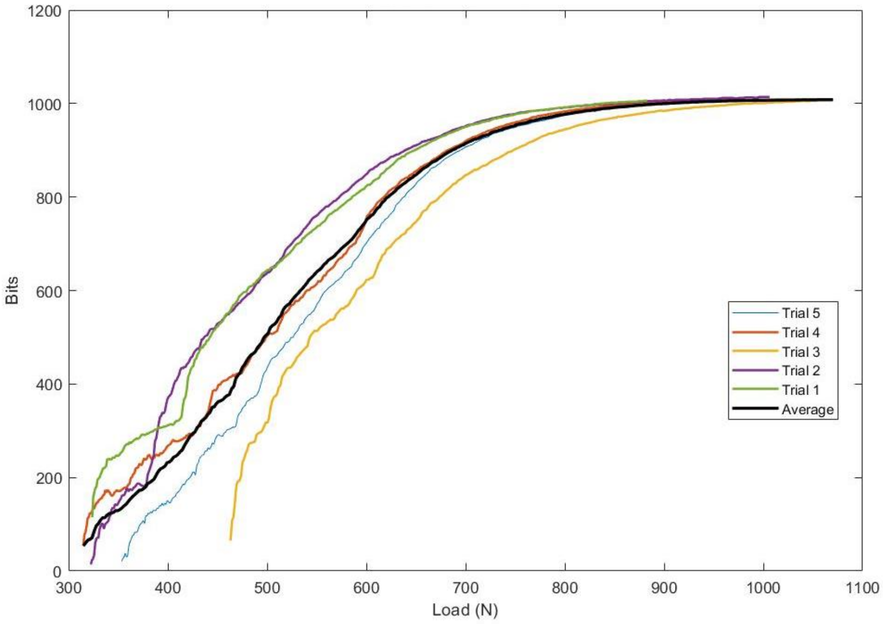


Figure 2.12. Bits vs. Load for 79mm of diameter. All five trials and their average are plotted.

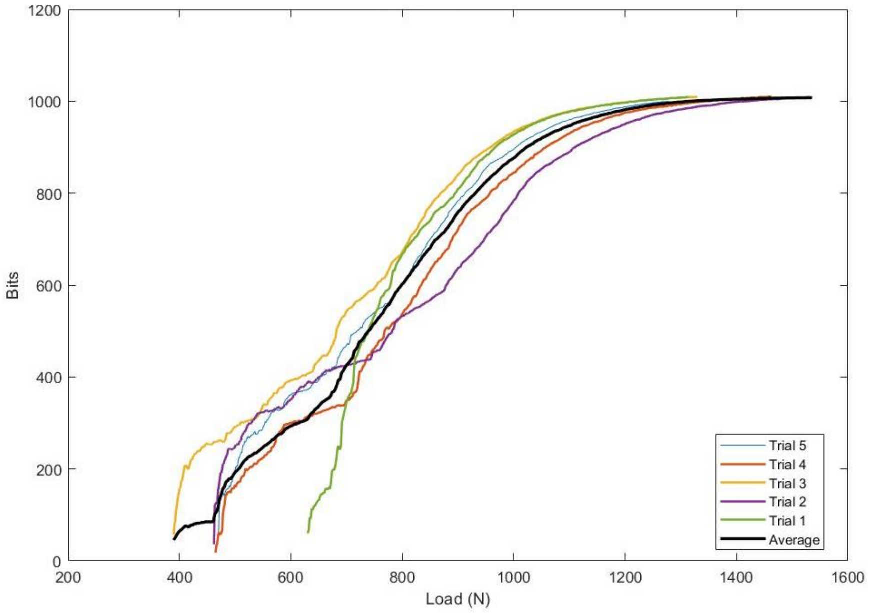


Figure 2.13. Bits vs. Load for 99 mm of diameter. All five trials and their average are plotted.

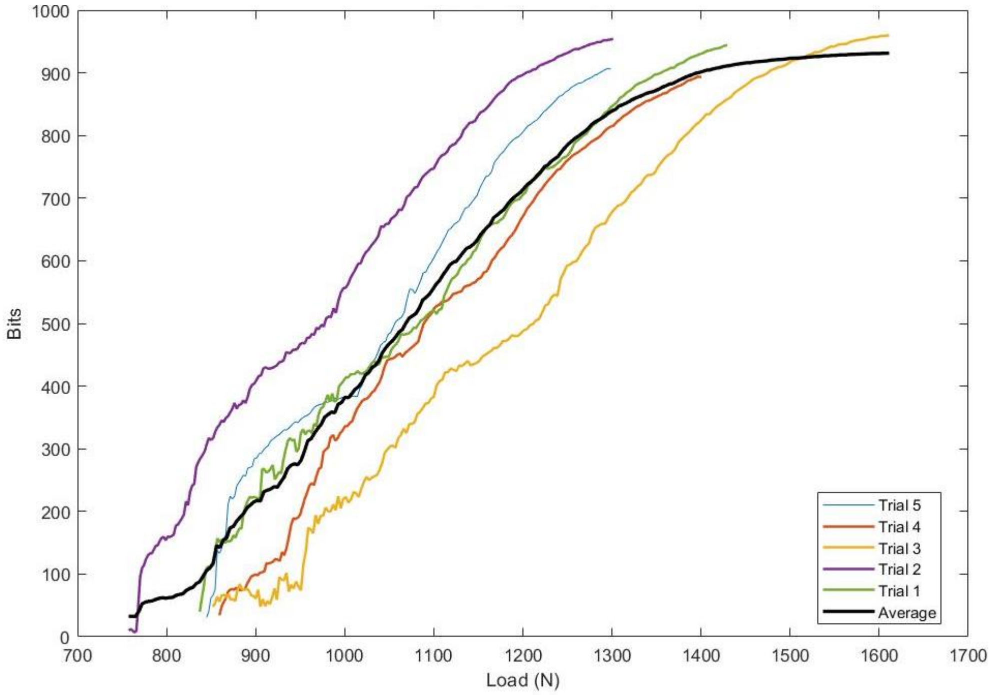


Figure 2.14. Bits vs. Load for 115 mm of diameter. All five trials and their average are plotted.

To find the relation between the bits and the load applied, the slopes of the linear part from each average plot were taken. The slope, in this case, represents the ratio between bits and load. To determine the slope, it was taken the first and last point of the linear behavior from each average curve:

$$\frac{bits}{load} = m = \frac{y_2 - y_1}{x_2 - x_1}$$

The slopes were plotted in function of the diameter to see which diameters of contact are the most suitable for the required application and how the ratio between bits and load (Figure 2.15). For diameters smaller than 80 mm, the ratio ramps up quickly and there is not a constant behavior. On the contrary, from 80 mm and above, the ratio between bits and load is linear and its behavior is close to the horizontal line, meaning that it is close to being constant.

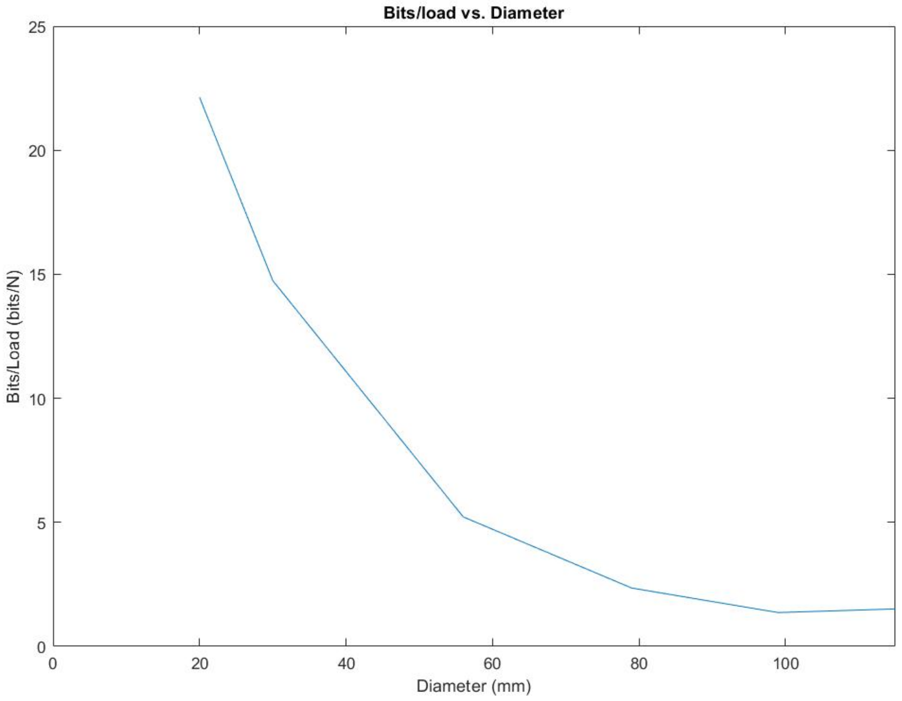


Figure 2.15. Bits/load in function of the diameter of the contact area.

Considering only diameters above 80 mm, linear fitting can be done (Figure 2.16). The load can be expressed as a function of the bits read by the Arduino micro-controller and the diameter of the contact area:

$$\frac{Bits}{load} = y = -0.024441D + 4.1285 \rightarrow \frac{[bit]}{[N]} = [mm]$$

$$\boldsymbol{Load = \frac{Bits}{-0.024441 * D + 4.1285}}$$

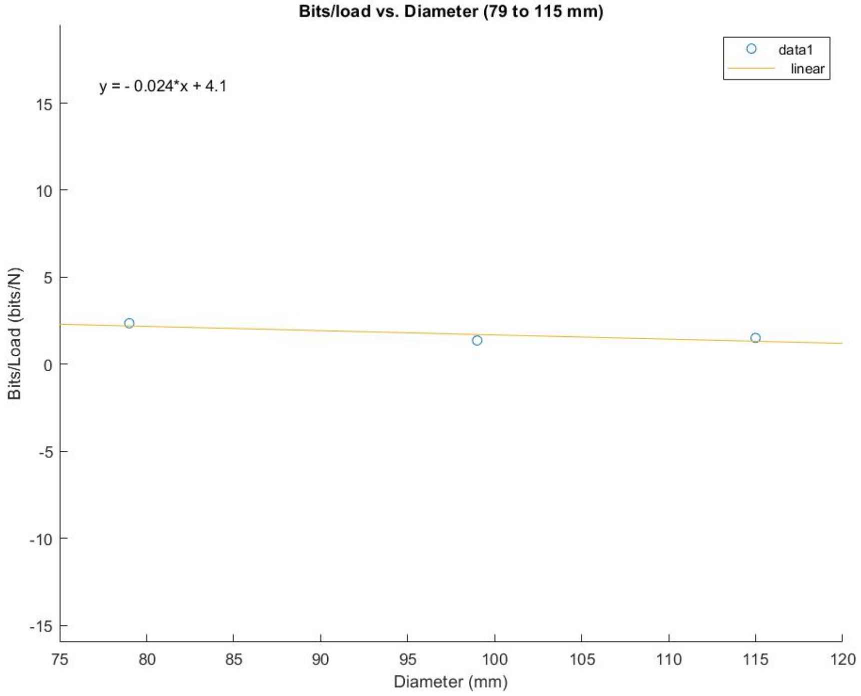


Figure 2.16. The linear part from the bits/load and diameter plot.

2.1.5. Final design

After processing the results from the calibration, it was decided that the size of the sensors will be two times the biggest dimension of the area of contact of Akwasi for each joint. In order to get an estimation of these dimensions from him, he used makeup on his shoulders and knees and placed his body on the sled on top of sheets of paper. Images of these prints were taken along with a measuring tape for scale (Figure 2.17). His right shoulder is 4.45 by 3.25 inches (11.30x8.25cm), his left shoulder had a similar size of contact area, 4.25 by 3.25 inches (10.79x8.25cm). His right knee is 3.25 by 2.5 inches (8.25x6.35 cm) and his left knee 3.50 by 2.75 inches (8.89x6.98 cm).

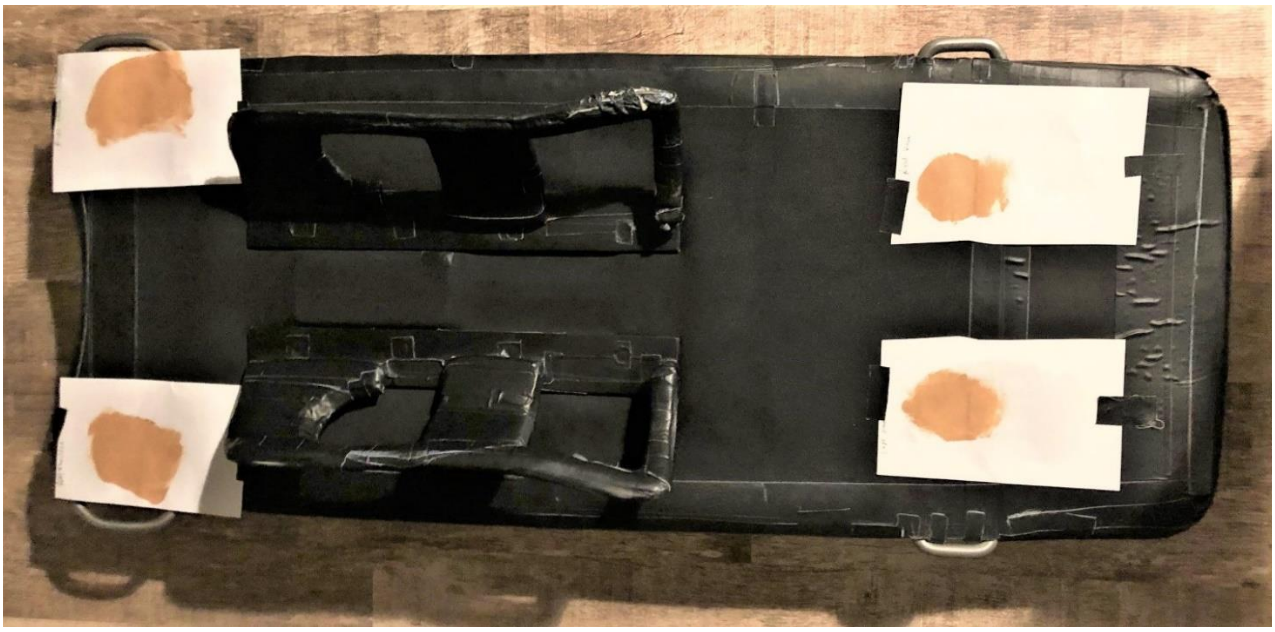


Figure 2.17. Area of contact prints from Akwasi. The bigger prints represent the shoulders and the smaller ones, the knees.

Considering his contact dimensions for shoulders and knees, the size for the sensors are:

- Shoulder steering force sensor: 20x20 cm
- Knee steering force sensor: 18x18 cm

These sensors were all manufactured by hand using a sewing machine Brother NV50, and the process is described in the following steps:

1. Make the templates for the microfiber fabric, conductive fabric, Velostat and mesh fabric as a spacer.
2. Cut all the fabrics, Velostat and spacer to the selected size.
3. Sew together the conductive fabric to the microfiber fabric (Figure 2.18).
4. Sew together the mesh fabric as spacer leaving one side open to place the Velostat inside (Figure 2.18).
5. Place each side of the sensor on top of each other, ensuring that the electrodes are in the right. position and the Velostat in the spacer are between them.
6. Sew the sensor together (Figure 2.19).
7. Sew the metallic snap buttons on each tab of the conductive fabric for each sensor with conductive thread, (Figure 2.19).

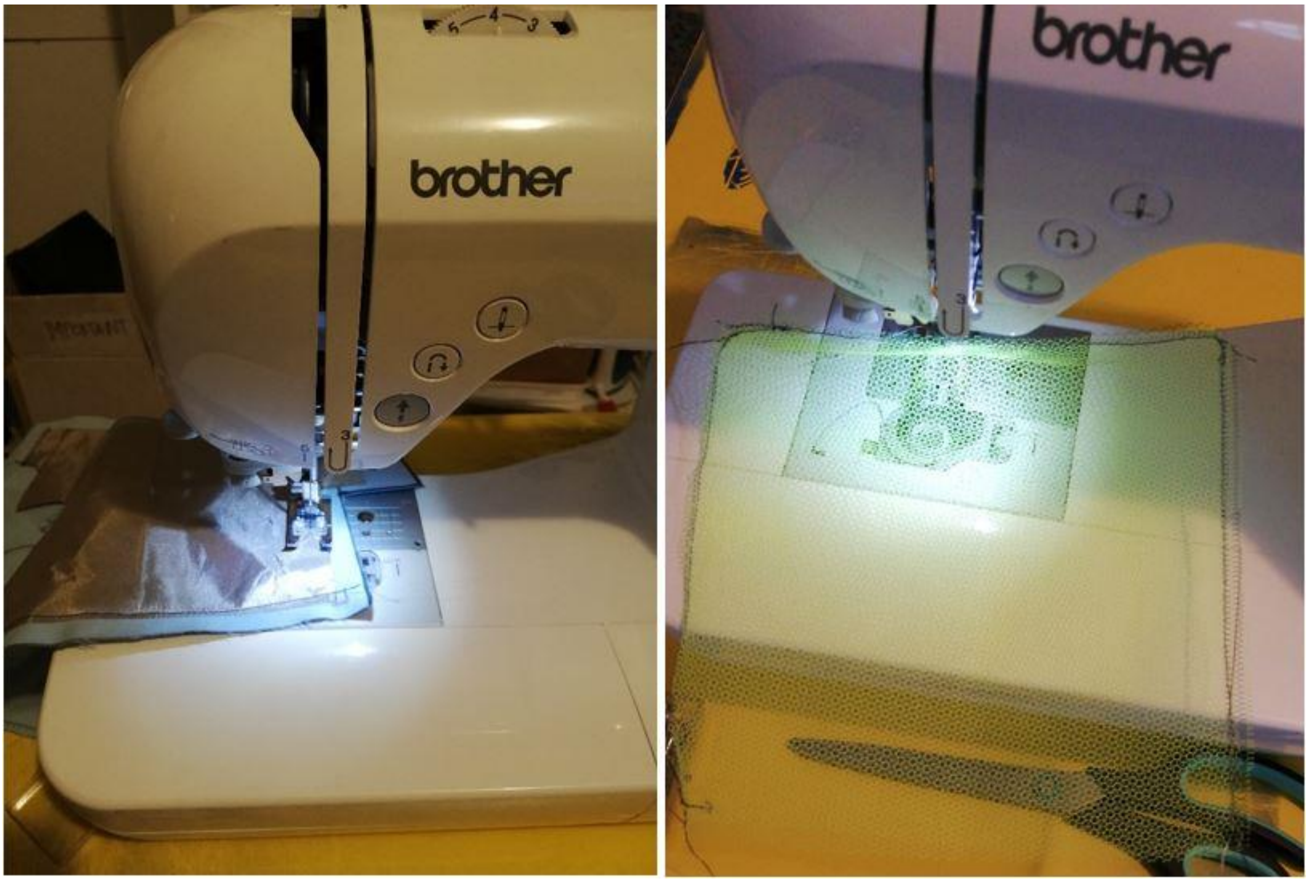


Figure 2.18. Manufacturing of the sensors. On the left, the conductive fabric is being sewed, the same for the spacer on the right picture.

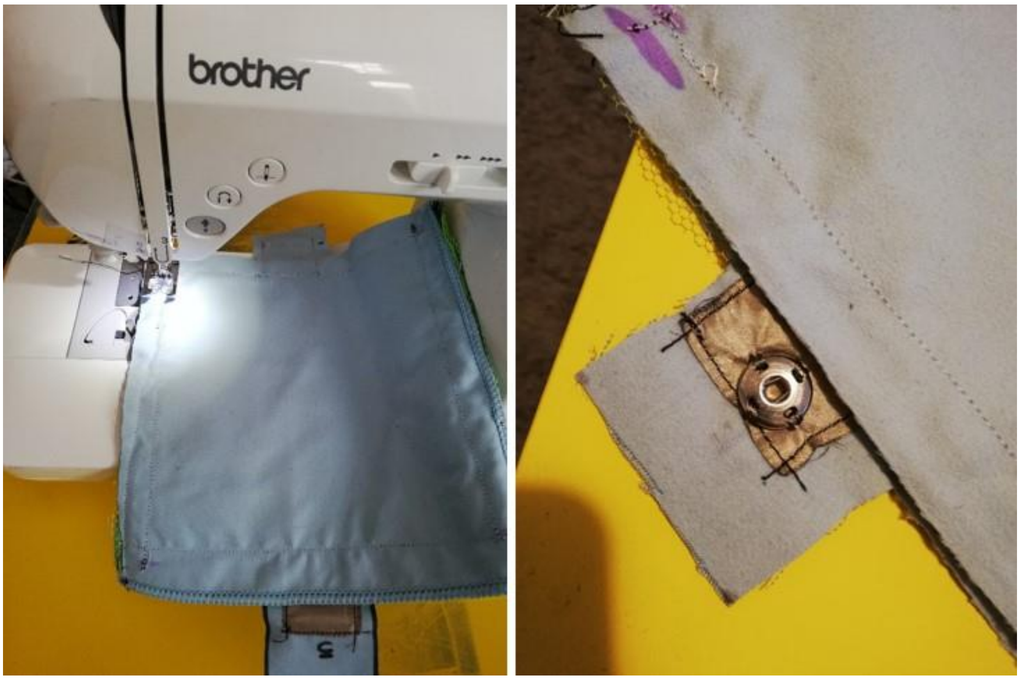


Figure 2.19. Last steps of the manufacturing of the sensors. In the left picture, the sensor is being sewed together and in the right one, the metallic button is attached.

After doing the previous manufacturing steps, a final version of the sensors was built (Figure 2.20).

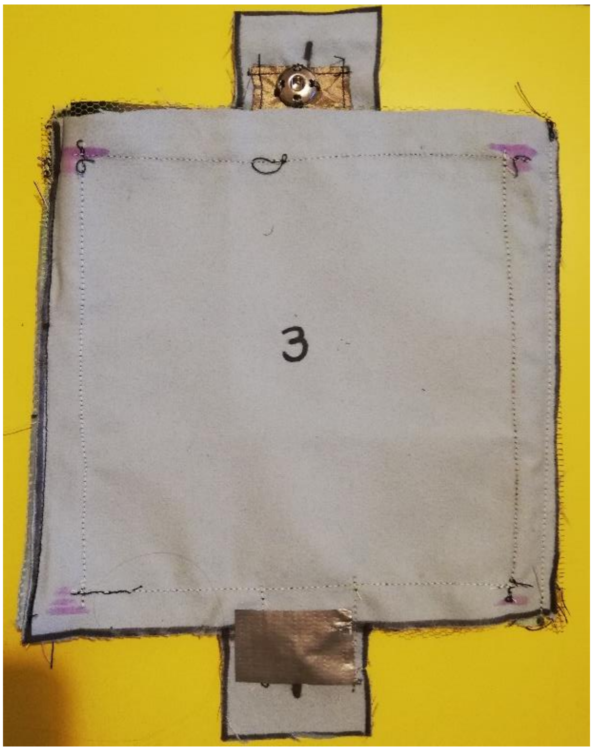


Figure 2.20. The final version of the steering force sensor.

2.2. Data collection and processing

Once the sensors have been developed, the next step is to connect them all together. These sensors are controlled through an Arduino Mega 2560. The circuit is very simple for each sensor: One end of the force sensor goes to 5 V pin in the Arduino. The other end is connected to Analog pin 0 and also to one end of a

100kΩ resistor. The other end of the resistor goes to GND on the Arduino. In this case, the resistor works as a pull-down resistor and a voltage divider, dividing the input voltage between the sensor and the resistor. Using a high resistor will improve the voltage range of the sensor making it more sensitive to small changes in the load applied. The end of the sensors was connected using snap-to-pin cables to secure the connection and jumper wires were used to connect the breadboard and the Arduino. A commercial Force Sensitive Resistor (FSR) was used to illustrate it since the sensor designed was not available [11]. (Figure 2.21)

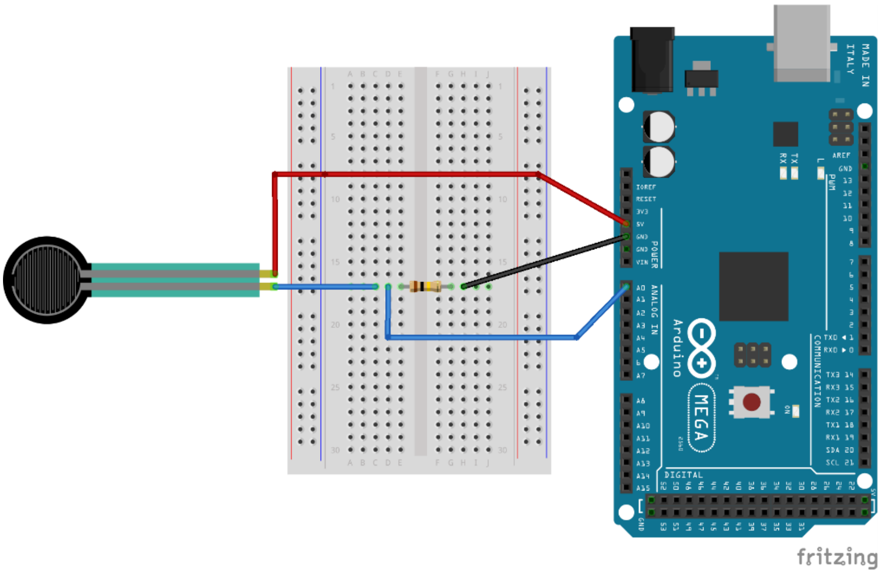


Figure 2.21. Schematic of the connection of the force sensor with the Arduino. An FSR is used for illustrative purposes, the designed sensor was not available.

An SD card shield was added to the Arduino in order to log the data collected during each run down the ice track (Figure 2.22). When the sensors are taken to the field, all the data collected during the practice runs will be stored in the SD card for further analysis.

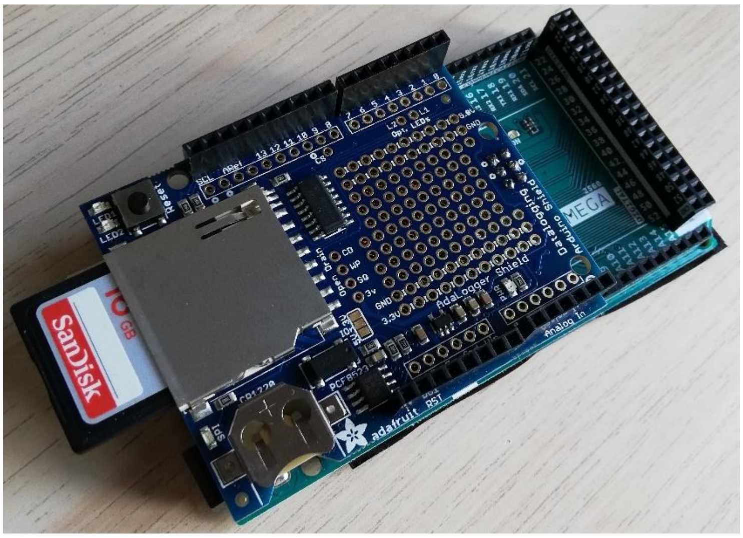


Figure 2.22. Arduino MEGA with the SD card Shield.

2.3. Placement on the sled

In order to place the sensors on the sled, some adjustments were made. The first step was to fill the spaces under the foam where the athlete lays. Since the frame of the sled is a rectangular steel structure, there are some empty spaces between the saddle and the frame. Because it is not a flat and regular surface, there is not enough support and the sensors cannot be placed. To fill these spaces, Styrofoam was used. With a metallic wire, it was possible to follow the complex shapes where the sensors will be placed. After having the contour of the required sections, the right patterns were carved in the Styrofoam to make it fit and make a smooth flat surface (Figure 2.23).

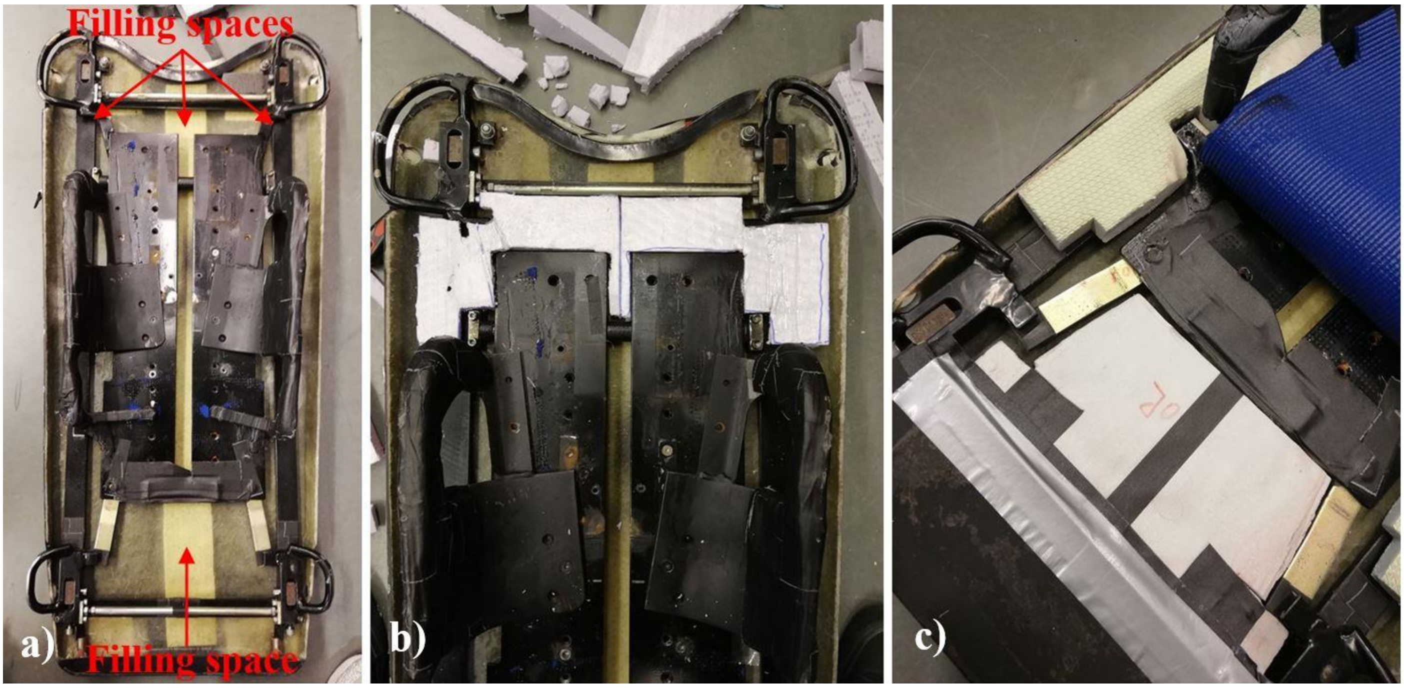


Figure 2.23. Filling the empty spaces in the sled. In figure a) the filling spaces on the sled are shown. In b) and c) is how the sled looks like after making the Styrofoam fillings in the top and bottom part of the sled respectively.

After filling the spaces, the following layers were added: a piece of foam mat on top of the Styrofoam sections added, the four sensors in the 4 different locations, and finally, the full foam mat covering the top of the sled. In this case, an additional piece of foam mat was added on the top and bottom of the sled in order to give even a smoother and flatter surface to place the sensors. The last layer of foam is used to give a soft surface to the sliders when they lie down on the sled and also to cover the sensors because by regulations nothing could be on top when the athletes are going down the track (Figure 2.24).

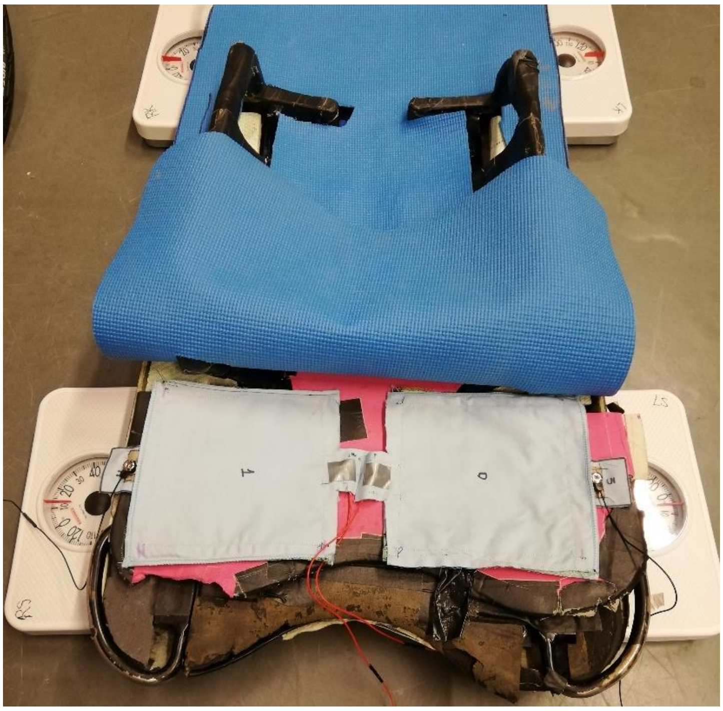


Figure 2.24. Placement of the sensors.

The pink foam is the first layer, then the force sensors are placed on top and lastly, the blue foam is placed on top of the sensors (Figure 2.24). To know where the shoulders and knees are located in order to place the sensors in the right place, Akwasi provided some pictures of him on the sled and highlighted were his shoulders and knees have contact with the sled (Figure 2.25).

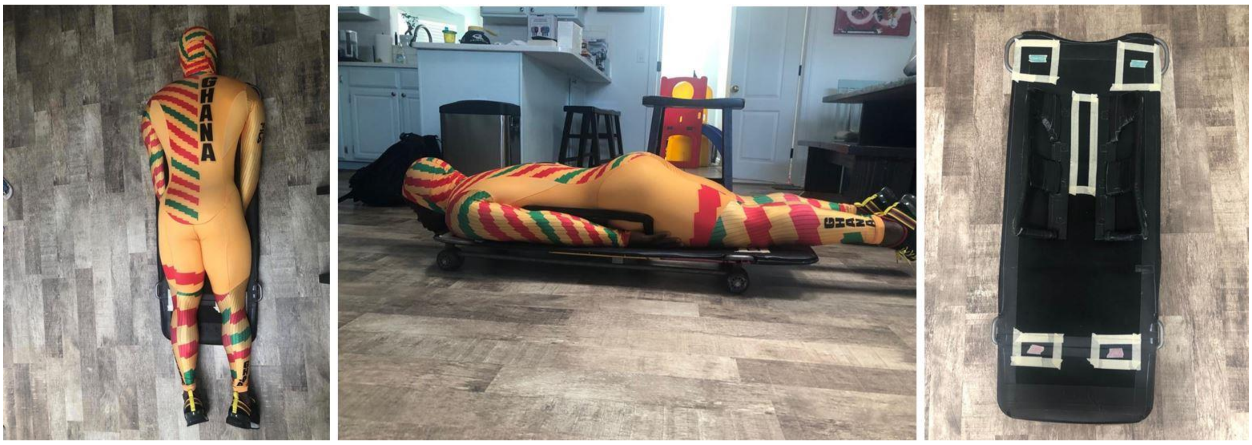


Figure 2.25. Akwasi Frimpong position on the sled. Shoulders and knees location to put the tactile force sensors.

3. Results

3.1. *Tests*

Before starting testing, it was important to find a subject with similar weight, height and body type as Akwasi Frimpong. He has a particular built, being 174 cm tall, around 88 kg and his body is very athletic. Fortunately, this subject was found, and his measurements were 177 cm tall and 87 kg. In addition, his shoulder and knee prints were taken in a similar procedure followed by Akwasi in order to be compared with Akwasi's measurements (Table 3.1).

Table 3.1. Comparison between shoulder and knee prints from Akwasi and the test subject.

Joint	Akwasi Frimpong	Test Subject
Left Shoulder (cm)	10.8 x 8.25	1.25 x 11
Right Shoulder (cm)	11.3 x 8.25	11.5 x 9
Left Knee (cm)	8.9 x 7	8.69 x 8
Right Knee (cm)	8.25 x 6.35	8.65 x 8

Measurements show a slight difference between the athlete and test subject as they have different bodies. The next step was to measure the weight distribution of shoulders and knees over the sensors by placing him over 4 scales (one for each joint) in the same position he will be laying down on the sled. The results are:

- Right Shoulder: 14 kg
- Left Shoulder: 12 kg
- Right Knee: 3 kg
- Left Knee: 3.5 kg

Regarding the setup, it was similar for all three tests. The sled was supported under 4 analog weight scales (one on each corner). These scales were used to read the load applied to each sensor. After supporting the sled on scales, the subject proceeded to lay down on it, on descent position, placing shoulders and knees on top of each sensor (Figure 3.1). These tests had to be in static conditions because it was not able to try the sensors on an ice track.

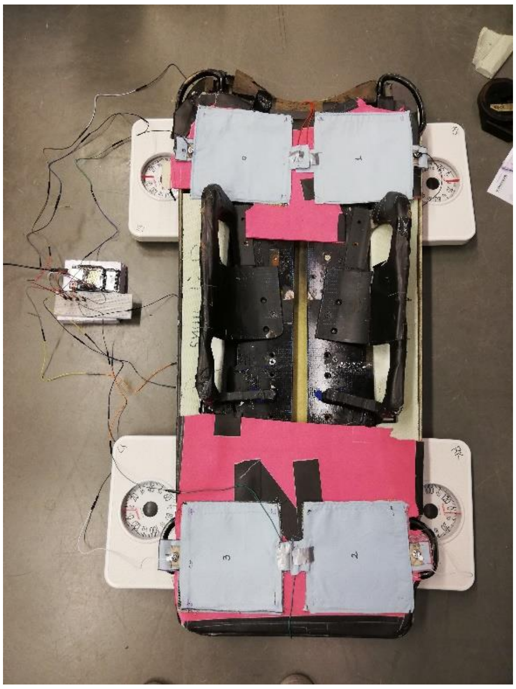


Figure 3.1 General set up for the tests.

The first test consisted of placing weights on top of each joint (one joint at a time) while leaving the other joints unloaded (Figure 3.2a). The readings on the scales and the response from the sensors were recorded. Increments of 2 kg were added until a maximum of 9 kg was reached. It was not possible going heavier than that because it was uncomfortable for the participant. This test was challenging to execute because placing and balancing weights on the same spot every time was difficult due to surface irregularity.

The second test consisted of applying a constant force to each joint (one at a time) while leaving the other joints without load and again register scales and sensors response. Periods of time, while each force was applied, were written down so it would be easier to recognize and analyze data. The increments were around 5kg each and it as carried until the subject reported discomfort. In this case, the test was more consistent and easier to accomplish because the person who was applying the force, kept his hands on the subject so the force was applied always on the same spot of each joint (Figure 3.2b).

The last test was more a validation test, the subject was told to simulate steering in skeleton by applying forces on the sensors with his joints, combining knees and shoulders, only shoulders or only knees (Figure 3.2c). The first two tests were to calibrate and create models (one for each sensor) to convert bits into force, and the last one to validate these models.

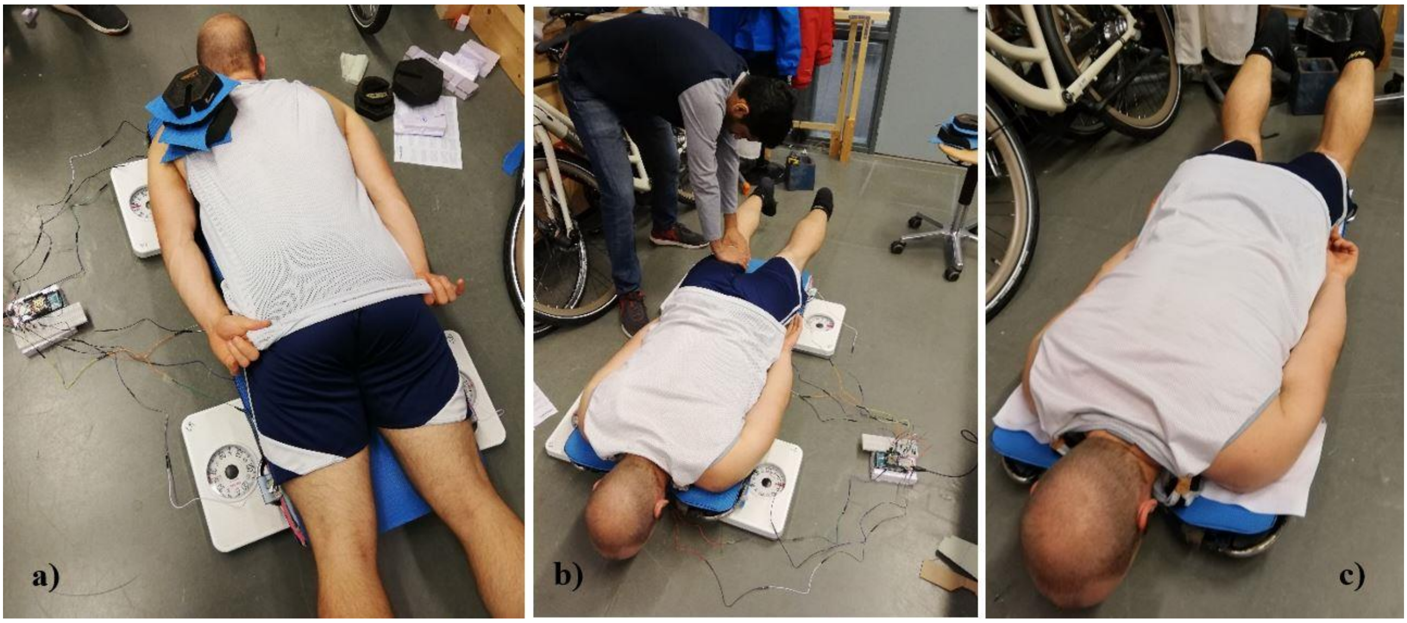


Figure 3.2. Tests accomplished. a) Fixed weights on top of each joint, b) Constant force applied by someone, c) Free steering forces applied.

3.2. Tests Results

After logging the results for each test on an SD card, the next step was to process the preliminary results and see how the sensors responded to different forces applied. The procedure to process data was similar for each sensor and test:

1) Highlight the periods of time where the forces were applied
2) Average the bits during these times
3) Plot the average bits with its respective force applied
4) Fit a linear regression for each sensor

In the following sections, these results and their trends are presented and illustrated. For the first test, only left shoulder and right knee results will be shown. The complete set of results for this test can be seen in Appendix A to C. All the results are presented from the second test are presented.

For further references, an “active sensor” refers to the sensor in which force is being applied. “Passive sensor” refers to sensors which remain unloaded.

3.2.1. Test 1 – Fixed Weights

Results from the first test showed mixing results. The test on the right knee (Yellow triangular markers; Figure 3.3). shows that the number of bits increased with the increasing force, having a linear behavior, where bits produced by the sensor increased proportionally to the force applied. Also, other sensors remained mostly constant, or with minimal changes, during this test.

However, other results were not as they were expected, that is to have a linear behavior. For example, the bits on the left shoulder sensor decreased when the force increased (Blue diamond markers; Figure 3.4) responded in the opposite way of what it was expected. In other cases, bits recorded in “passive” sensors in which force is not applied (Right shoulder and left and right knee) remain mostly constant (Figure 3.4). This was expected because no loads were applied to these joints.

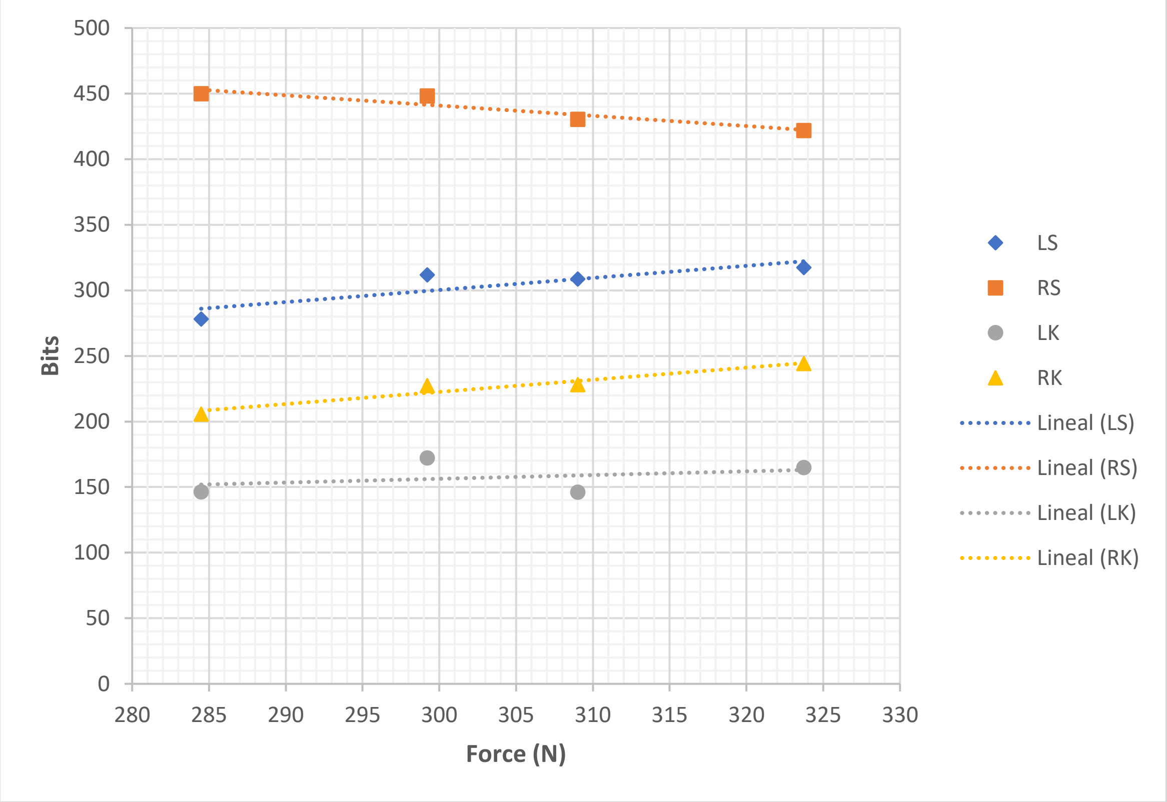


Figure 3.3. Test 1. Bits vs. Force: Right Knee

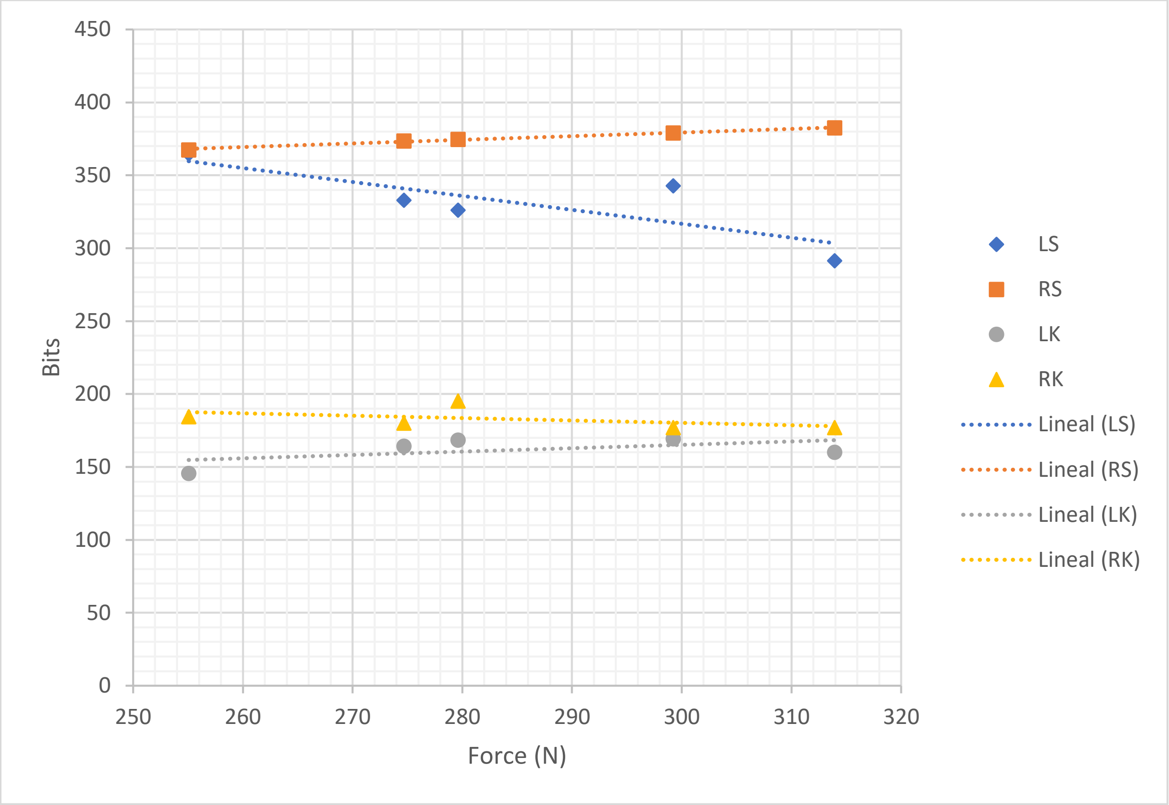


Figure 3.4. Test 1. Bits vs. Force: Left Shoulder.

3.2.2. Test 2 – Constant Force

The second test was closer to the reality of how the athlete applies steering forces on the sled. Results were more consistent among the four sensors and their response to the applied forces was as expected; A linear behavior between the number of bits and force applied on the “active” sensor and almost constant values for the other three “passive” sensors (Figure 3.5 to Figure 3.8)

The number of bits increased when more force was applied. Since only one joint was pressed at the same time, all other sensors should give a constant number of bits during the whole trial. Three trials were conducted for each sensor. Results from each sensor have some variations between trails. However, results follow a linear trend and the bits and force are proportional (Figure 3.5 and Figure 3.8). Only one result from the first trial is shown, all other results can be found in Appendix B.

Comparing the results between tests 1 and 2, changes in bits when the force increase is more notorious and marked in test 2 (Figure 3.5 and Figure 3.8). In addition, sensors that were not pressed remained more constant and stable than test 1.

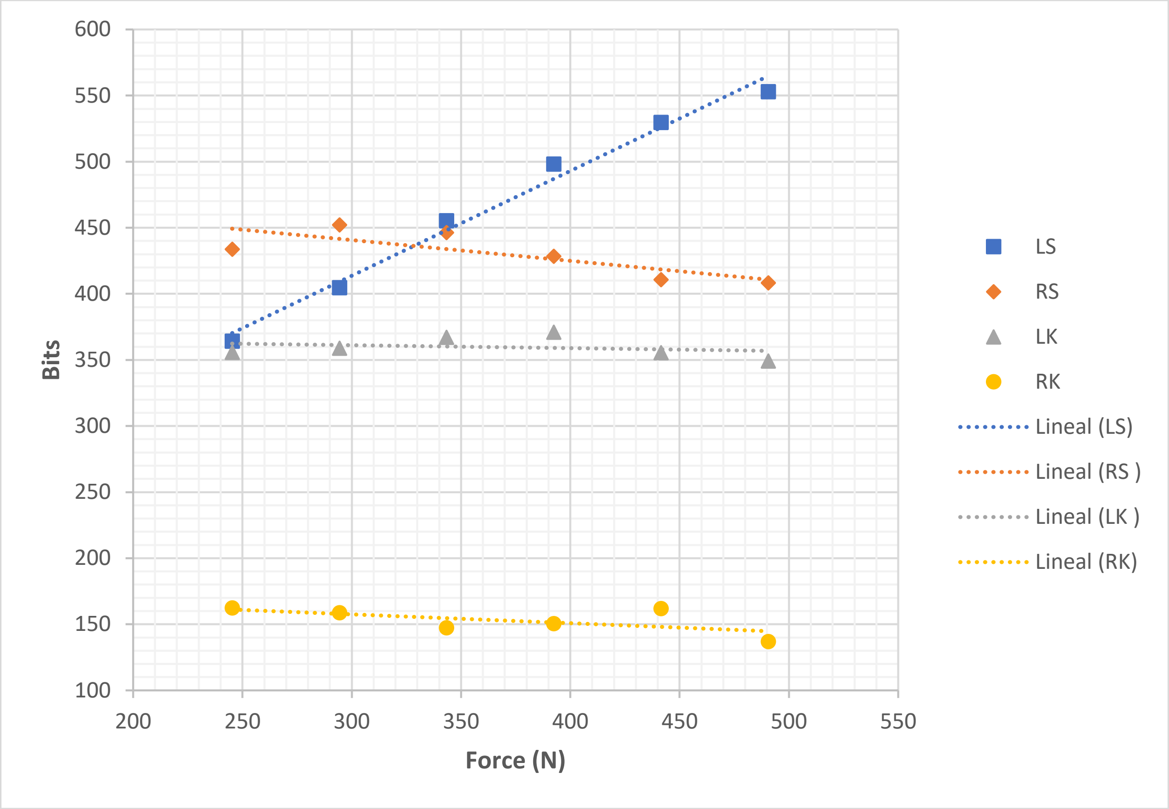


Figure 3.5. Test 2. Bits vs. Force: Left Shoulder – trial 1

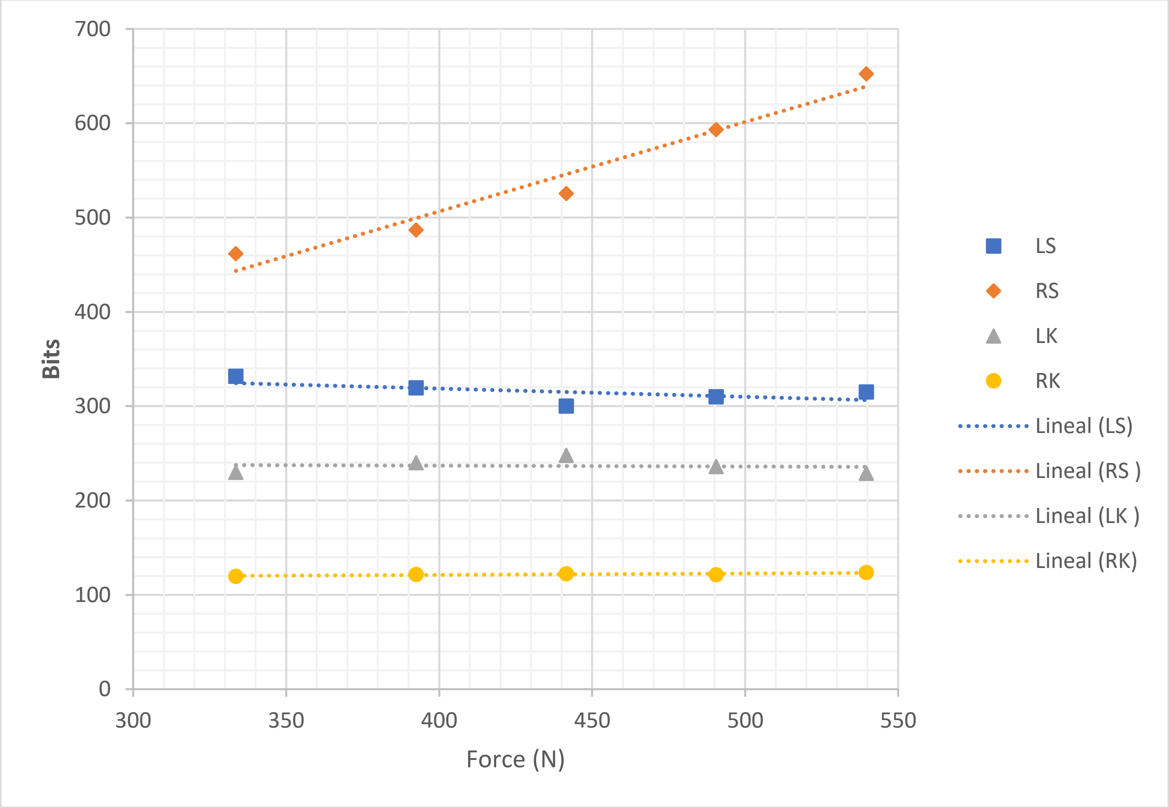


Figure 3.6. Test 2. Bits vs. Force: Right Shoulder - Trial 1

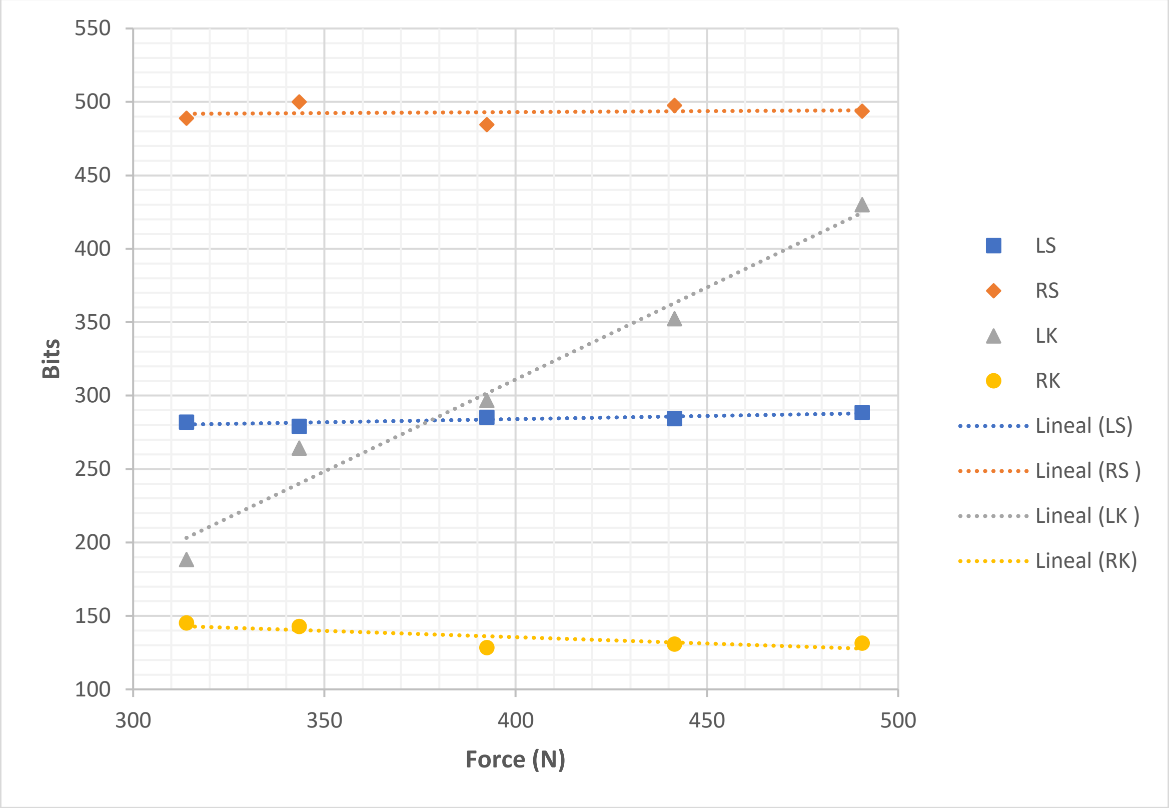


Figure 3.7. Test 2. Bits vs. Force: Left Knee -Trial 1

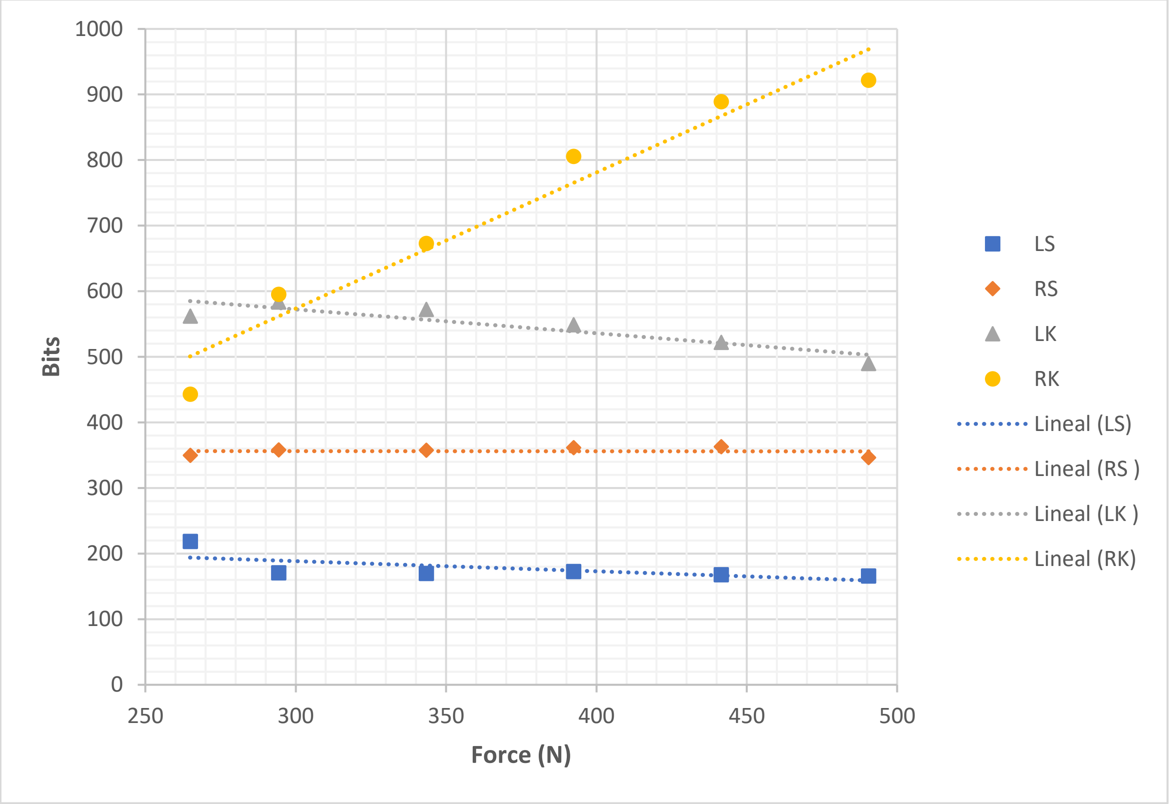


Figure 3.8. Test 2. Bits vs. Force: Right Knee

3.3. Steering Force Models

Once all results were processed, a model to convert bits into force was created. Since each sensor was built by hand, there are some small differences between them. For this reason, it was decided that each model will have its own steering force model. Since three trials were executed and results between them were different, an average of the results was considered to find each model. Error bars were included in the graph to show the variability of the results between trials.

After obtaining the models, validation tests were executed to see how well each model works. These tests consisted of letting the subject apply forces in a similar way that a skeleton athlete steers down the ice track. The loads shown on the scales and the periods of time were recorded for further comparisons with the models' results.

3.3.1. Left Shoulder Model

Once the results from test 2 were processed, a linear regression of average bits was fitted to get the model for the left shoulder sensor. The error bars show the maximum and minimum bits recorded after three rounds (Figure 3.9).

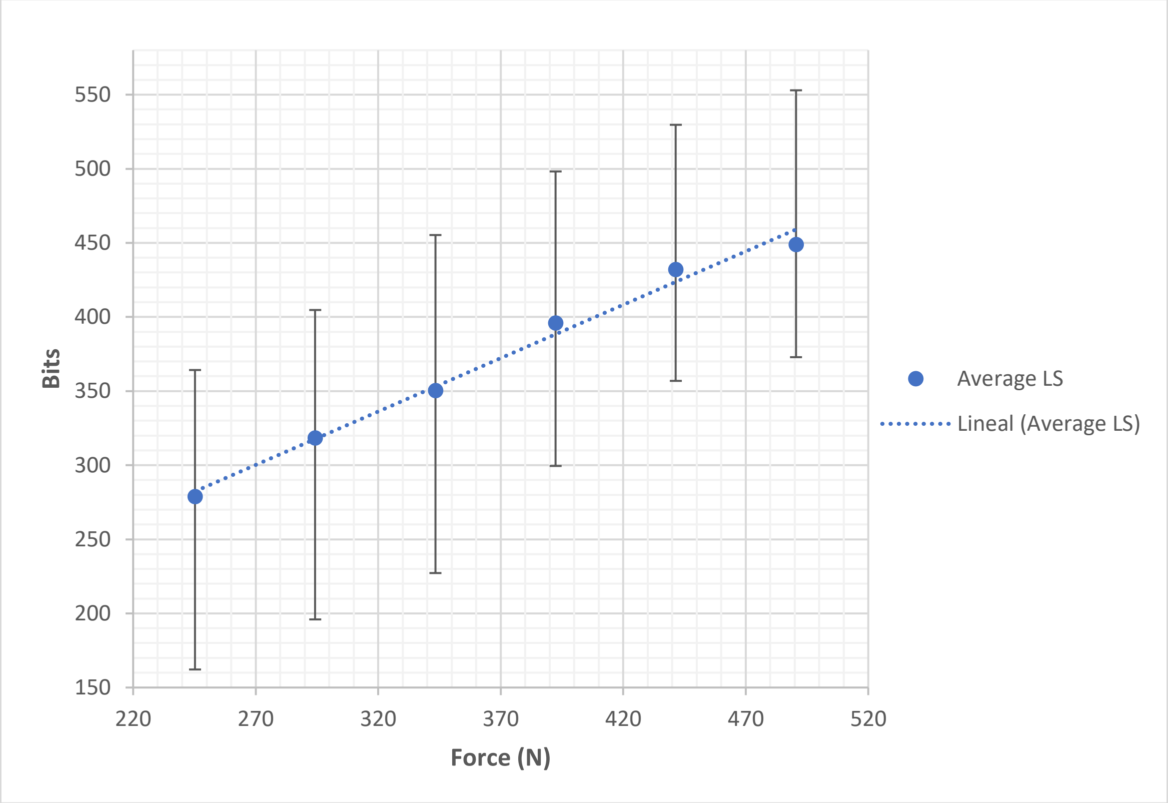


Figure 3.9. Linear regression for left shoulder sensor.

The equation shows the relationship between bits recorded by the sensor and the force applied by the subject (in Newtons):

$$Bits = 0.7205 * F_{LS} + 105.66$$

$$\boldsymbol{F_{LS}} = \frac{Bits - 105.66}{0.705}$$

3.3.2. Right Shoulder Model

The same procedure from the left shoulder was applied to the right shoulder sensor, fitting a linear regression to the average of bits recorded. Error bars show the maximum and minimum bits recorded after the third trial (Figure 3.10).

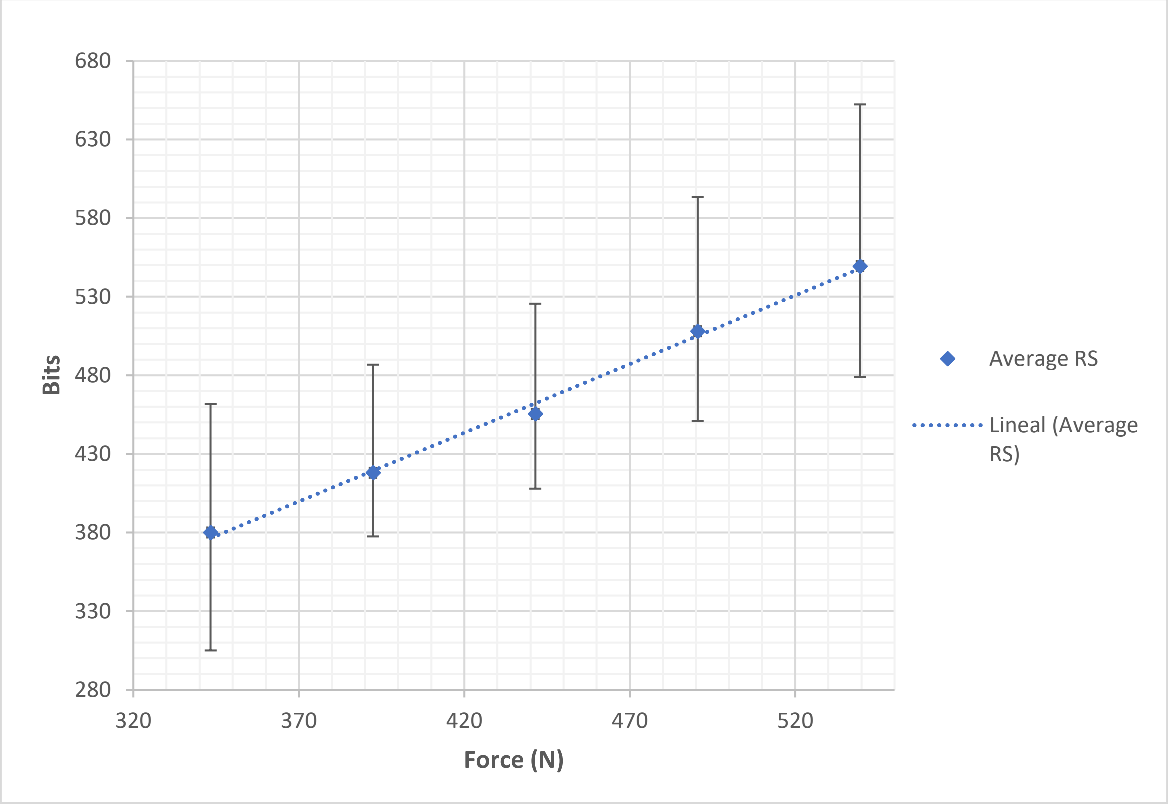


Figure 3.10. Linear regression for right shoulder sensor.

The following equation shows the relationship between bits recorded by the sensor and the force applied by the subject (in Newtons):

$$Bits = 0.8739 * F_{RS} + 76.434$$

$$\boldsymbol{F_{RS}} = \frac{Bits - 76.434}{0.8739}$$

3.3.3. Left Knee Model

The procedure was the same for the left knee sensor model. A linear regression was fitted into the average bits and the equation of that line is the model for this specific sensor. Error bars show the maximum and minimum bits recorded after the third trial (Figure 3.11).

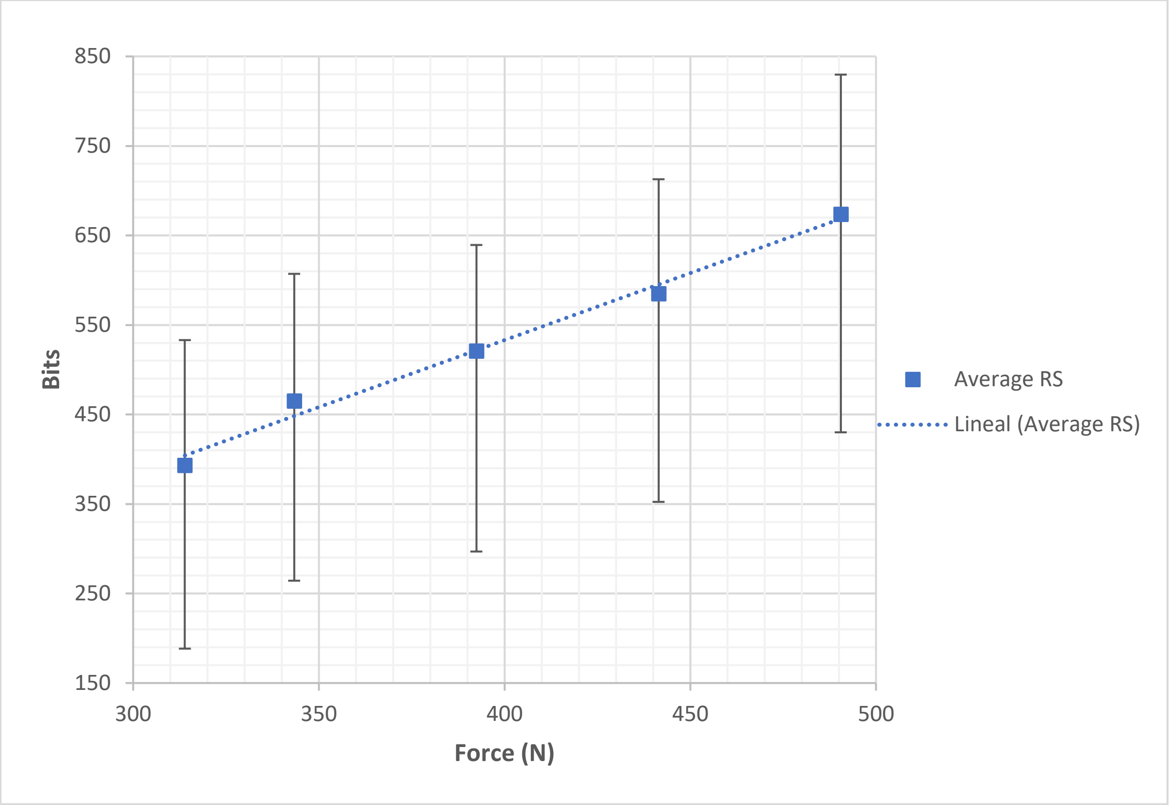


Figure 3.11. Linear regression for left knee sensor.

The following equation shows the relationship between bits recorded by the sensor and the force applied by the subject (in Newtons):

$$Bits = 1.4972 * F_{LK} - 65.828$$

$$\boldsymbol{F_{LK}} = \frac{Bits + 65.828}{1.4972}$$

3.3.4. Right Knee Model

Finally, average bits from the third trial for the right knee sensor were fitted into a linear regression. The error bars show the maximum and minimum bits recorded after three runs (Figure 3.12).

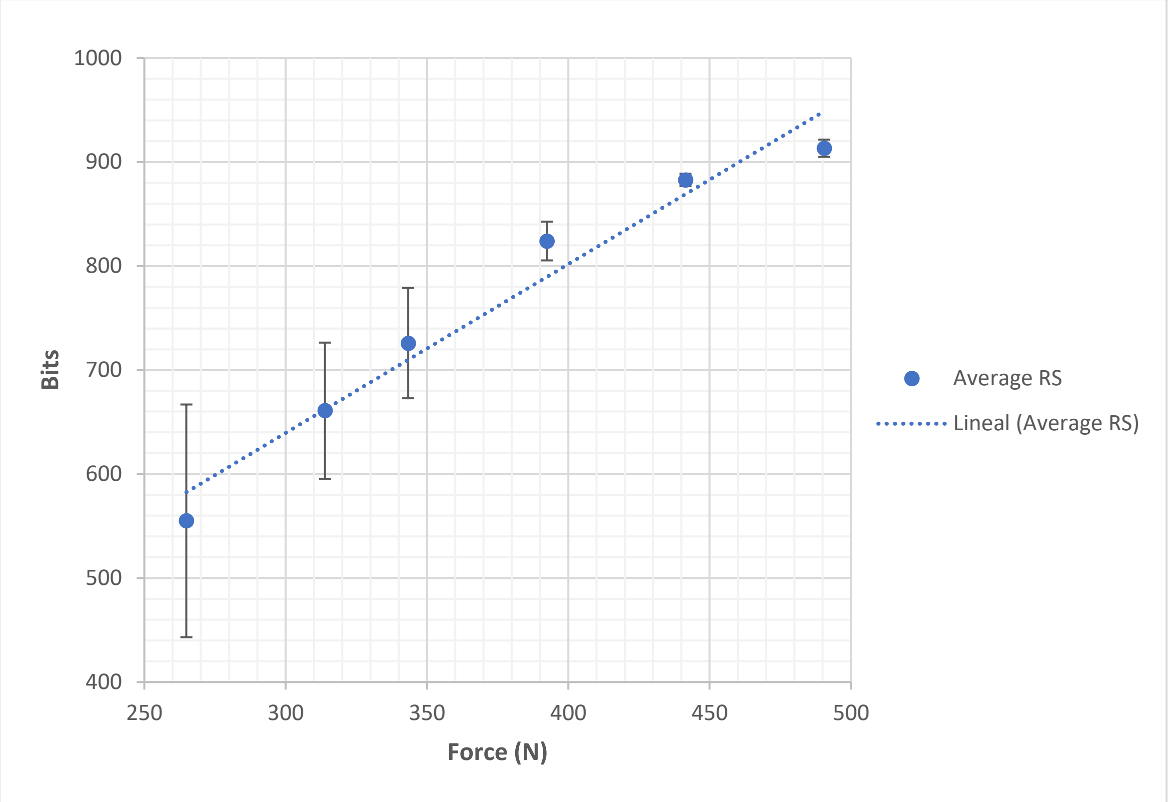


Figure 3.12. Linear regression for right knee sensor.

The following equation shows the relationship between bits recorded by the sensor and the force applied by the subject (in Newtons):

$$Bits = 1.6237 * F_{RK} + 152.41$$

$$\boldsymbol{F_{RK}} = \frac{Bits - 152.41}{1.6237}$$

3.3.5. Validation test – Free steering Forces

Once the models to convert bits into force were found for each sensor, validation tests were executed. In these tests, the subject was asked to steer the sled applying force in the following order:

1. Left shoulder + right knee
2. Right shoulder + left knee
3. Left shoulder
4. Right shoulder
5. Left knee
6. Right knee

The subject had to hold a force while the load on the scales was recorded. These periods of time were also registered in order to recognize when force was applied. Three trials were performed. Comparisons between the load on the scales and the ones calculated with the models are presented below (Figure 3.13 to Figure 3.16). To illustrate the comparisons, the reading of bits was average during the duration for which corresponding forces were applied. The load applied and calculated were plotted against time. Only the first trial results are shown, the other two trials can be found in Appendix C.

.

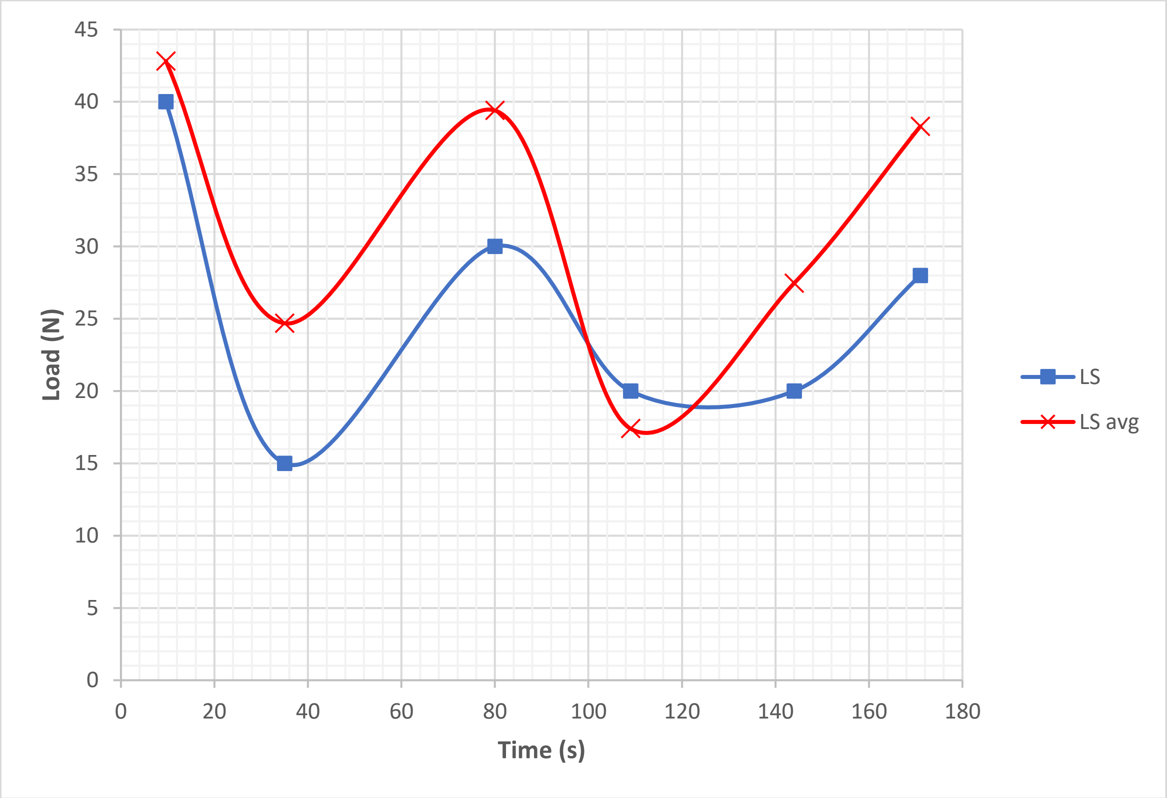


Figure 3.13. Free steering test, trial 1. Left shoulder sensor. The blue line (squared markers) represents the load applied and the red line ('x' markers) the calculated load with averaged bits.

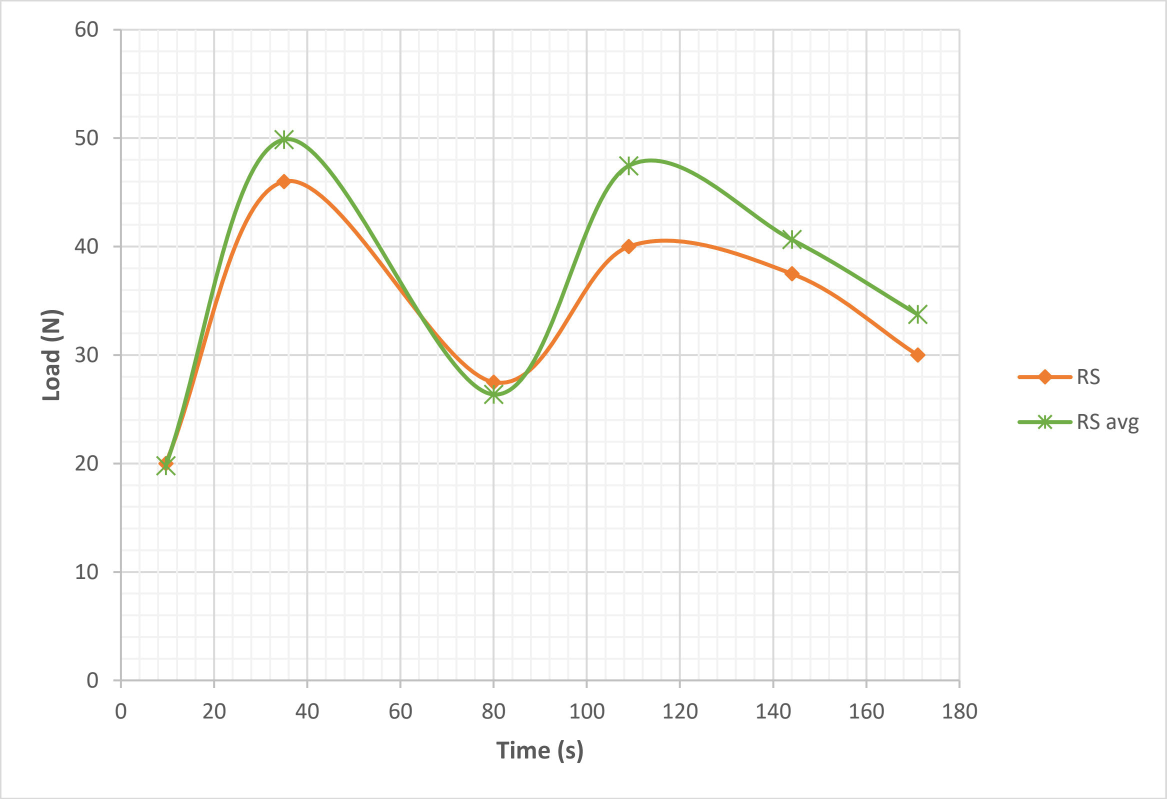


Figure 3.14. Free steering test, trial 1. Right shoulder sensors. The orange line (diamond markers) represents the load applied and green line (star markers) the calculated load with averaged bits.

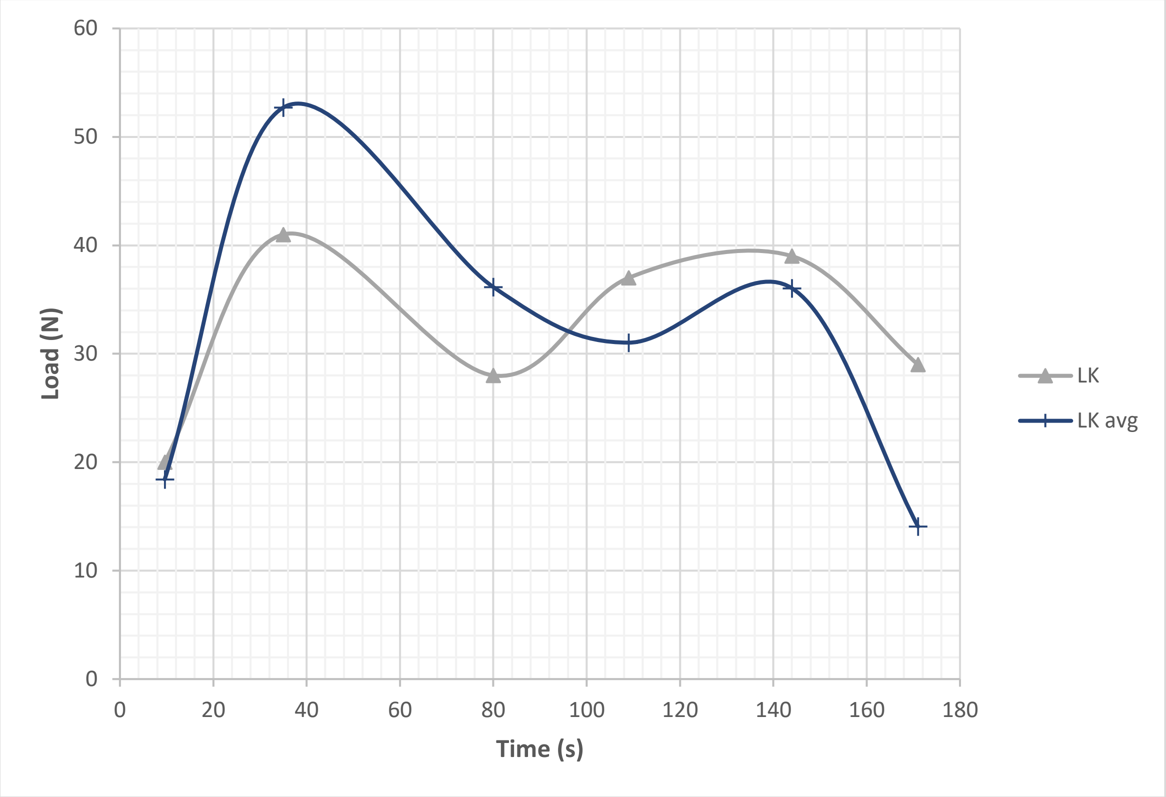


Figure 3.15. Free steering test, trial 1. Left knee sensors. Grey line (triangle markers) represents the load applied and the dark blue line (cross markers) the calculated load with averaged bits.

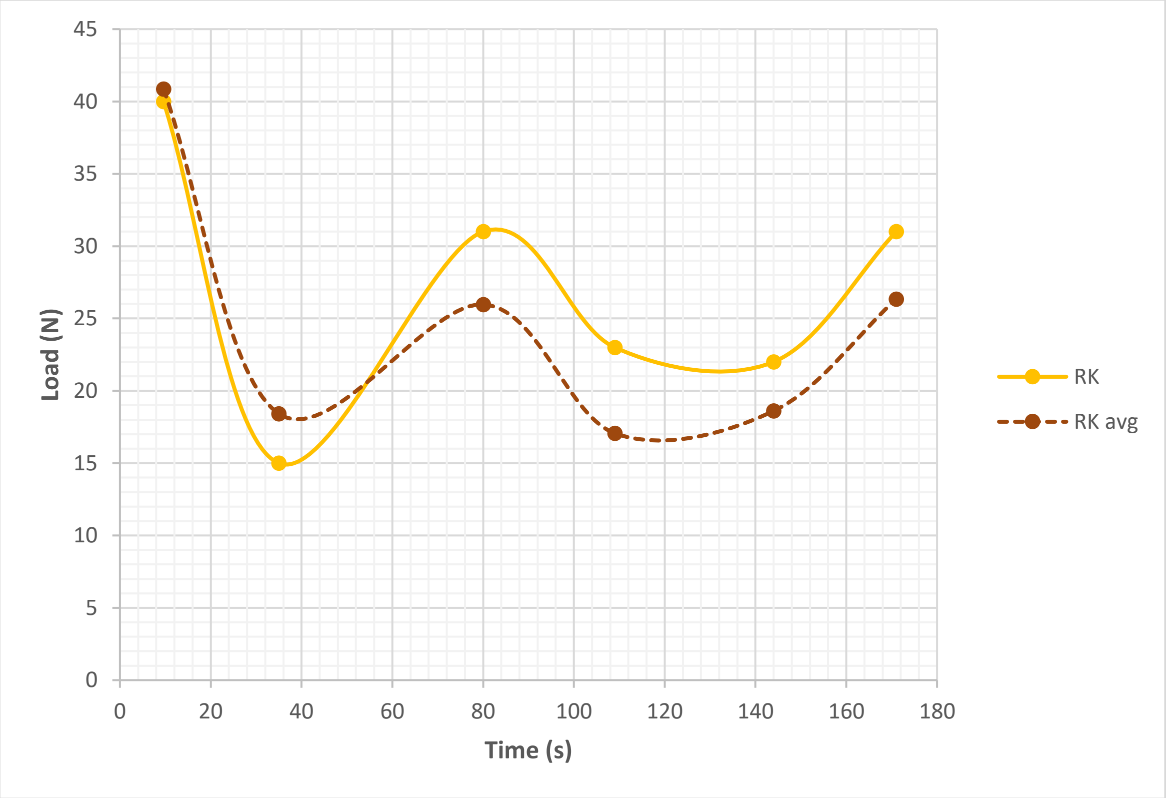
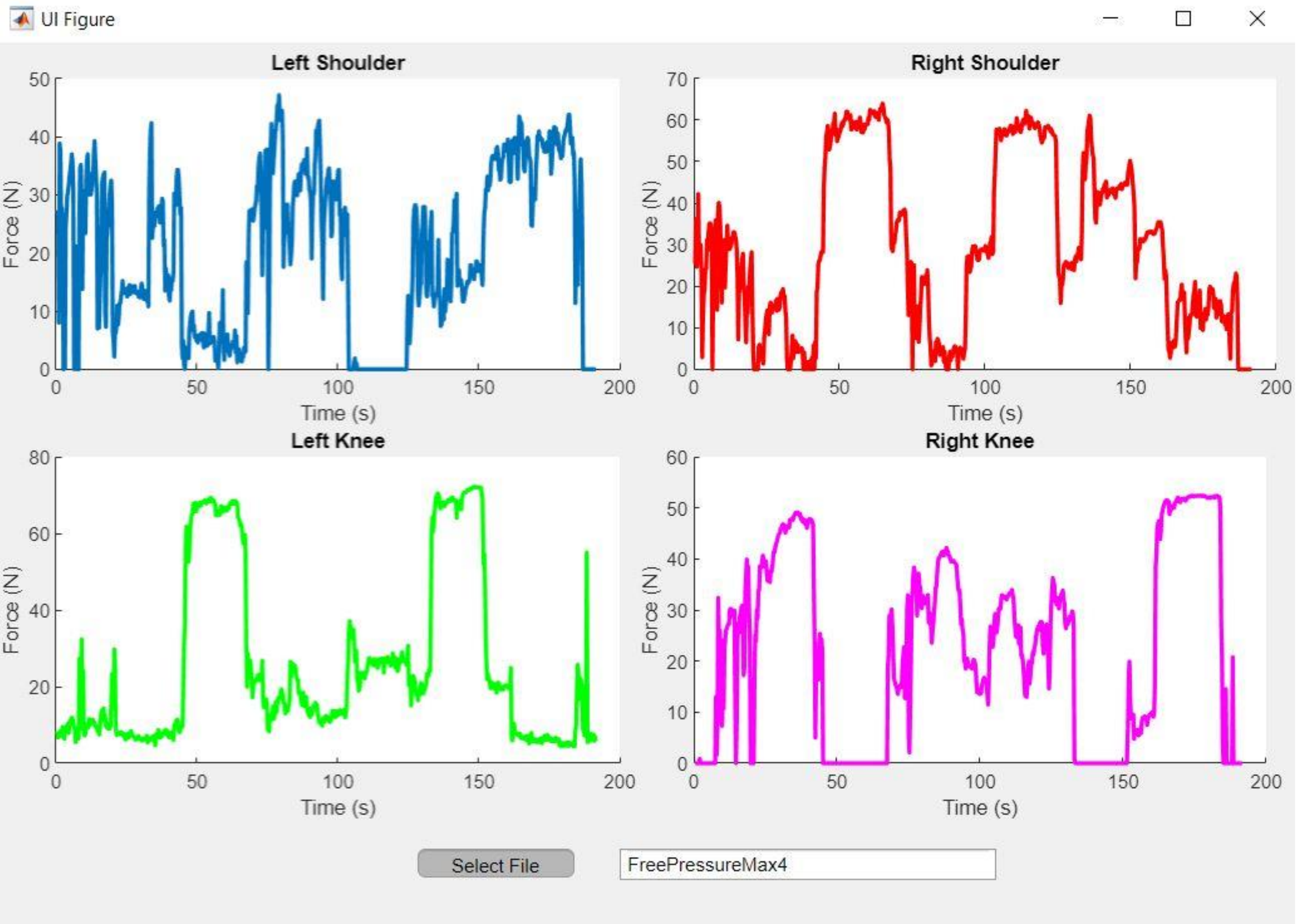


Figure 3.16. Free steering test, trial 1. Right knee sensors. Yellow line (round markers) represents the load applied and brown line (dotted line) the calculated load with averaged bits.

3.4. *Presentation to the athlete*

Once the data of the descend is collected, the athlete needs to see the results of his performance in a simple and easy way to read them. To accomplish this, a graphical user interface (GUI) was developed (Figure 3.17). This interface allows the athlete to select a file from the SD card and plot the results recorded by each sensor.



FreePressureMax4

Figure 3.17. Steering Forces interface.

4. Analysis and Discussion

4.1. *Analysis of the Steering Force Models*

The free steering results (Figure 3.13 to Figure 3.16) showed that the sensors have good responses and behavior during these validation tests. They showed that the calculated load follows a trend and have similar behavior compared to what actual values were. It is true that there is an offset in numbers but the sensors performed close to what they should be (Figure 3.13 to Figure 3.16).

The coupled steering joints (left shoulder + right knee and right) as well show similar behavior in their trends, both having the peaks and valleys at the same time. Furthermore, it can be recognized that even when only joint was pressing, its counterpart was involved and helping to steer (Figure 4.1 and Figure 4.2). For example, in the third data point (around 80 s) of Figure 4.1, only the left shoulder was applying force reaching a high load, but at the same time, the right knee sensor recorded an increase in the force applied. This shows that it is hard for an athlete to decouple from its opposite side joint, and always there will be an involuntary contribution from both joints at the moment of steering.

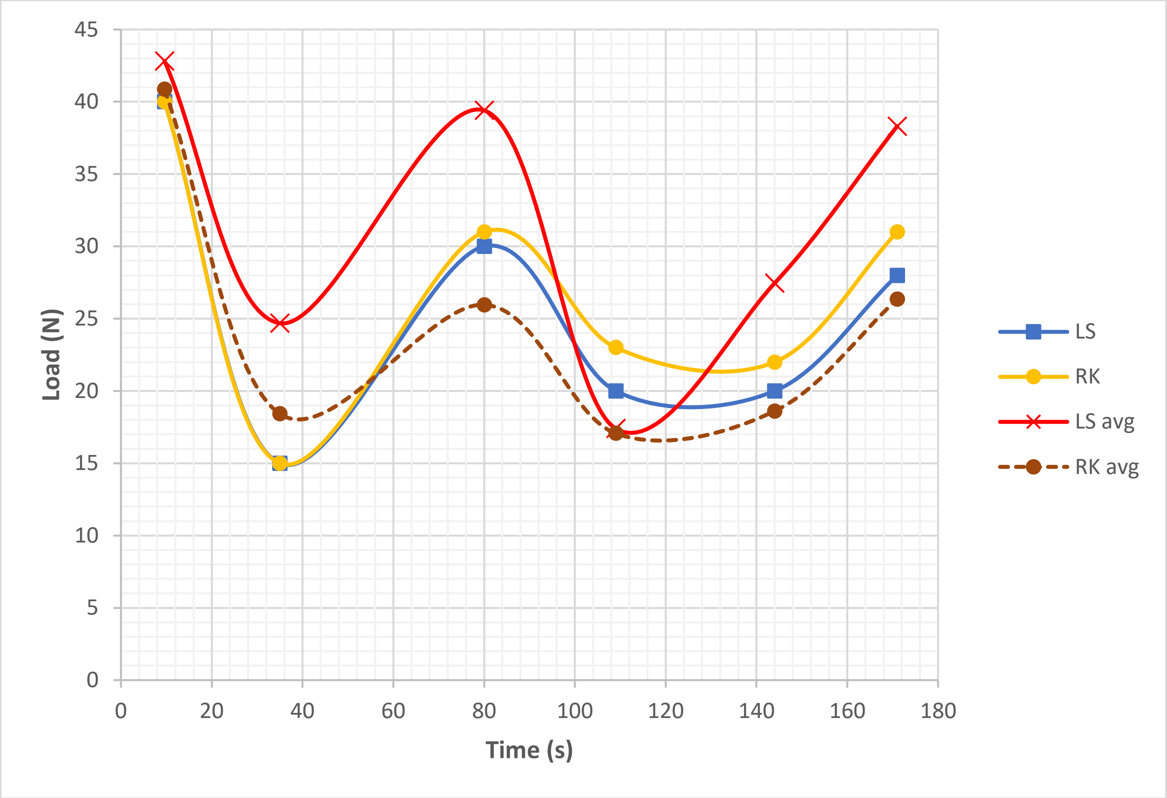


Figure 4.1. Free Steering Test trial 1. Blue (squared markers) and yellow (circular markers) line correspond to the load recorded from the scales for the left shoulder and right knee. Red ('x' markers) and the brown (dotted line) lines correspond to the load calculated with the models. It can be seen that the calculated load follows the trend of the applied load with offsets in some data points. Also, when only the left shoulder or right knee is applying force, the other joint also has an increase in the force.

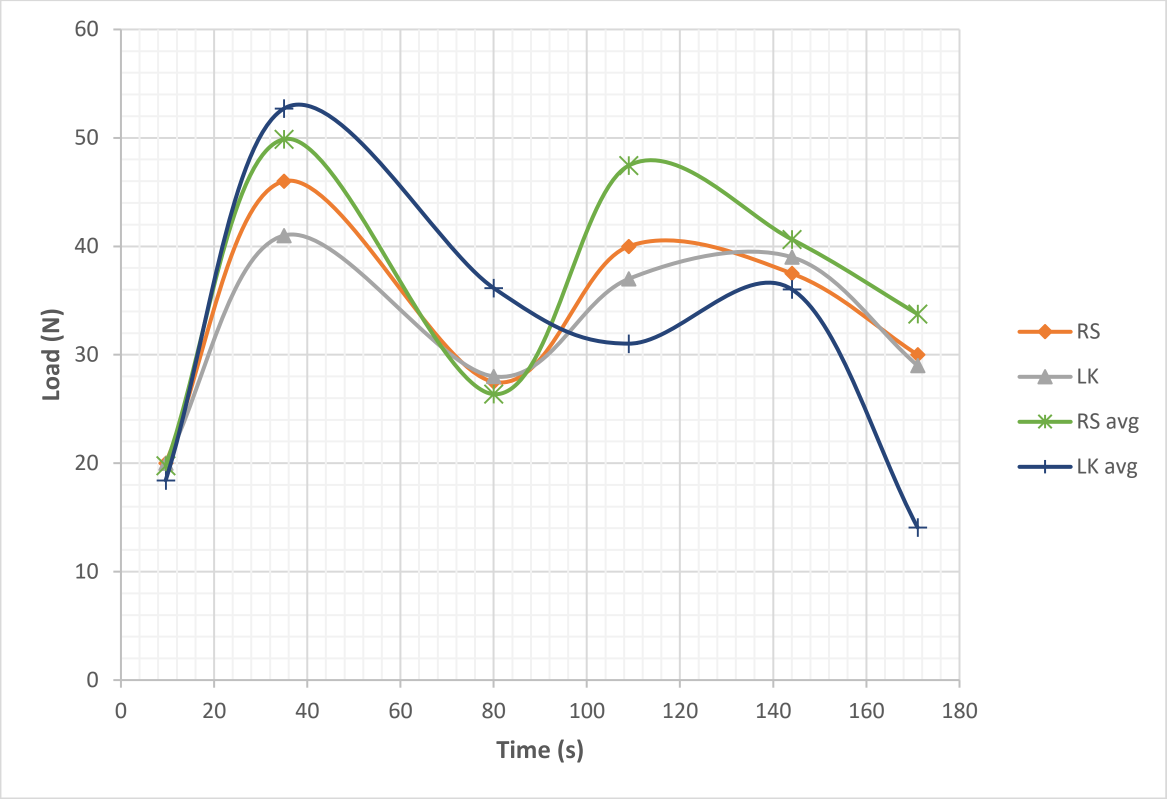


Figure 4.2. Free Steering Test trial 1. Orange (diamond markers) and grey (triangular markers) lines correspond to the load recorded from the scales for the right shoulder and left knee. Green (star markers) and dark blue (cross markers) lines correspond to the load calculated with the models. It can be seen that the calculated load follows the trend of the applied load with offsets in some data points. Also, when only the right shoulder or left knee is applying force, the other joint also has an increase in the force.

However, the results (Figure 3.13 to Figure 3.16) shows that there is a difference between the force applied by the subject and the one calculated with the models. These differences can be attributed to different factors: 1) the sled used, was for a shorter athlete (less than 175 cm) but the subject selected was too tall for the sled. This means that the placement of sensors needed to be moved a little bit up and down to be under the shoulders and knees accordingly. This affected the recordings of bits because the sensors were not fully supported by a flat and smooth surface but by a harder and more irregular one. Therefore, when the subject applied force to steer, the reaction forces were higher and, as a result, a higher number of bits.

2) Although sensors were made following the exact same procedure, there are some differences between them. For example, the conductive fabrics could be in slightly different positions or piezoresistive material sizes could be different. Also, when sewed together the two layers an initial offset pressure on the sensors could be added. That is why it was expected that each sensor has its own model and a different range of bits in which they operate. 3) Lastly, since the subject had to hold the force applied to steer the sled, the load recorded from scales was oscillating during each trial. Making difficult to know exactly what the exact force was applied.

4.2. Accelerometers Integration with the Steering force sensors

Besides steering force sensors, inertial measurements are also beneficial to the athlete. For this reason, the use of accelerometers could provide this information. In this case, the idea is to combine previously developed force sensors and an inertial measurement unit (IMU). The IMU will be a Shimmer from Real-Time Technologies. The Shimmer is a small wireless sensor platform with integrated kinematic sensors, large storage and low-power consumption that allows motion capture, data acquisition and real-time monitoring [12]. The main components of the device consisted of two triaxial accelerometers, one triaxial gyroscope, one triaxial magnetometer, a MicroSD card slot for the data logging and Bluetooth connection (Figure 4.3).

Figure 4.3. Shimmer3. [13]

Since this is an independent system with its own data logging system (through a microSD card), there are not many options to connect it with steering force sensors developed. However, the manufacturer offers different software that can be used to stream, visualize and process collected data. Within the options there is a software called Consensys, drivers to use the sensor with MATLAB or an Android application. Even though real-time streaming could be very useful for coaches to see athlete's performance, in this case, it is not a viable option because the connection would be lost as soon as the athlete starts moving away from the coach. The best option is to save the data collected in an SD card, similar to the force sensors and processed it after each run or at the end of the training.

Data from both sensors can be time-matched during the data processing stage by performing an abrupt motion that perturbates the sensors. The greatest option to achieve this is when the athlete jumps into the sled before start descending. In this case, both the shimmer and steering force sensors will show a high spike at the same time. These spikes will be the cue to match both systems' data.

Data matching will allow an athlete to see the force applied during the steering and the accelerations and g-forces developed. The following steps should be done to get data recorded by both sensor systems: 1) Both sensors have to be turned on to start recording data. 2) When the athlete performs the loading on the sled, sensors will record that impact. 3) When the athlete reaches the track end, both sensor systems must be turned off to stop collecting data. Finally, both SD cards must be removed and plug them in the pc to start the visualization and processing of data.

5. Conclusion

The main goal of this project was to instrument a skeleton sled by developing a measuring system that allows an athlete to understand better his performance down the ice track. Four tactile flexible force sensors were built to measure the steering forces of an athlete. It is based on the principle of piezoresistive effect and it measures applied force by measuring changes in resistance of piezoresistive material. The sensors have linear behavior between bits recorded and forces applied, showing that change in resistance is proportional to change in force. After data were collected through an Arduino and logged into and SD card, four models (one for each sensor) were created to convert bits into force. These models calculated the force properly and the results obtained followed the same tendency of real force applied.

However, calculated forces with the models were not exactly the same as the forces applied. This offset could be attributed to the handmade manufacturing process of sensors and the placement of them on the sled. Since sensors are developed for Akwasi to use, a participant with similar physical attributes was used for testing. The sled was provided by the Dutch Sliding Federation was too small for the subject and that could contribute additional errors in measurements because the positions of the joints on the sled were not in the right spots, making subject press the sensors in the wrong position.

In addition to force sensors, a shimmer will be included to measure the accelerations down the ice track. Unfortunately, the force sensors and the shimmers cannot be synchronized directly because the latter has its own data acquisition system. For this reason, the coupling of these two systems must be done during data processing. An idea was proposed to know the exact time when both sensors start measuring and know when to start matching data from both systems. This time cue must be a perturbation where both sensors record a high peak in the data. This can be achieved during the loading moment when the athlete jumps into the sled after the push start.

Finally, to present results to the athlete was proposed the use of a simple Graphical User Interface (GUI) application where an athlete can see results and his performance down the track. The development of these sensors could give Akwasi great insights on which weak parts of his sliding he can target and improve, helping him better understand his interaction with the track to obtain better results during his competitions.

5.1 Future Work

The future work should focus on optimize and compress the electronics of force sensors in order to fit them inside a tight space in the sled. At the moment, Arduino, SD card shield and all the wires are taking a lot of space that can be hindering athlete's performance. Moreover, a real-time clock could be added in order to register the date and time of each run to keep track of his performances.

Additionally, all four sensors could be done with a more standardized process making them all equal. This will reduce errors that handmade manufacturing process could add. In addition, the GUI app could be improved by making it more interactive allowing the athlete and their coach to see and recognize easier areas where the athlete can address and improve. Finally, once the electronics have been compressed, the force sensors should be tested on a real run down the ice track.

Appendix A – Fixed Weights

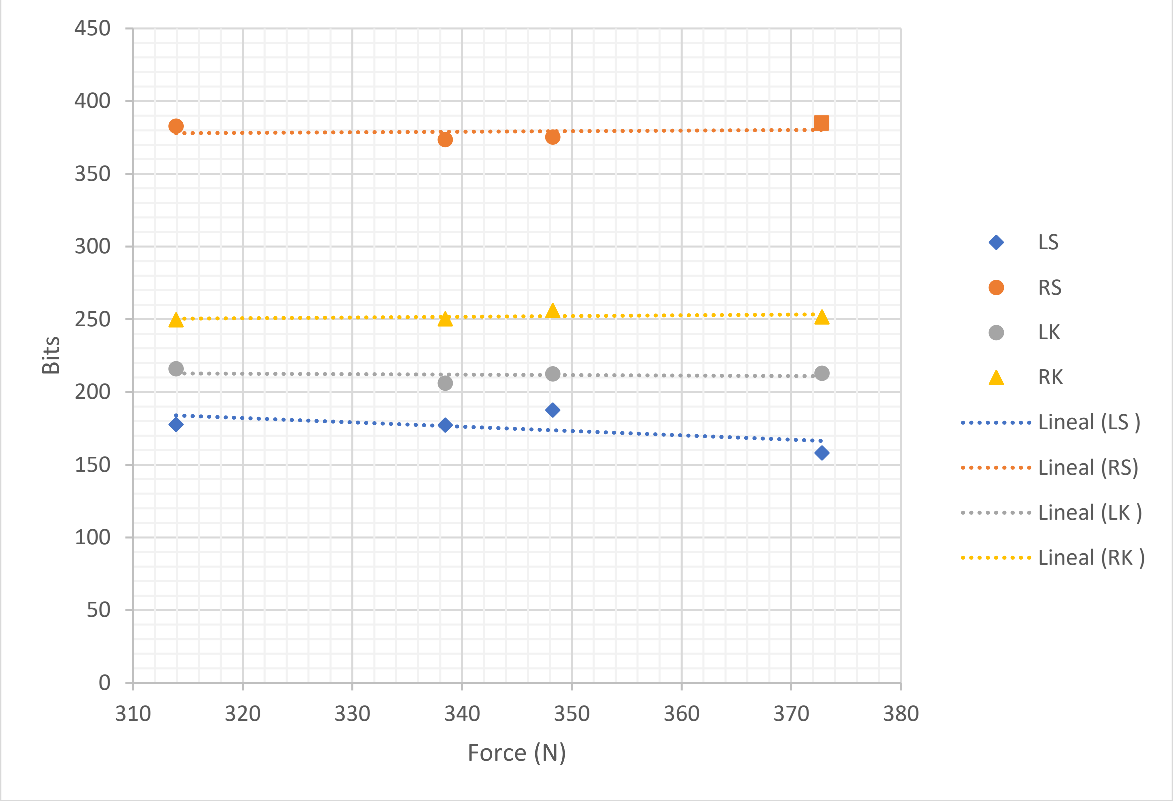


Figure A.1. Test 1. Bits vs. Force: Right Shoulder.

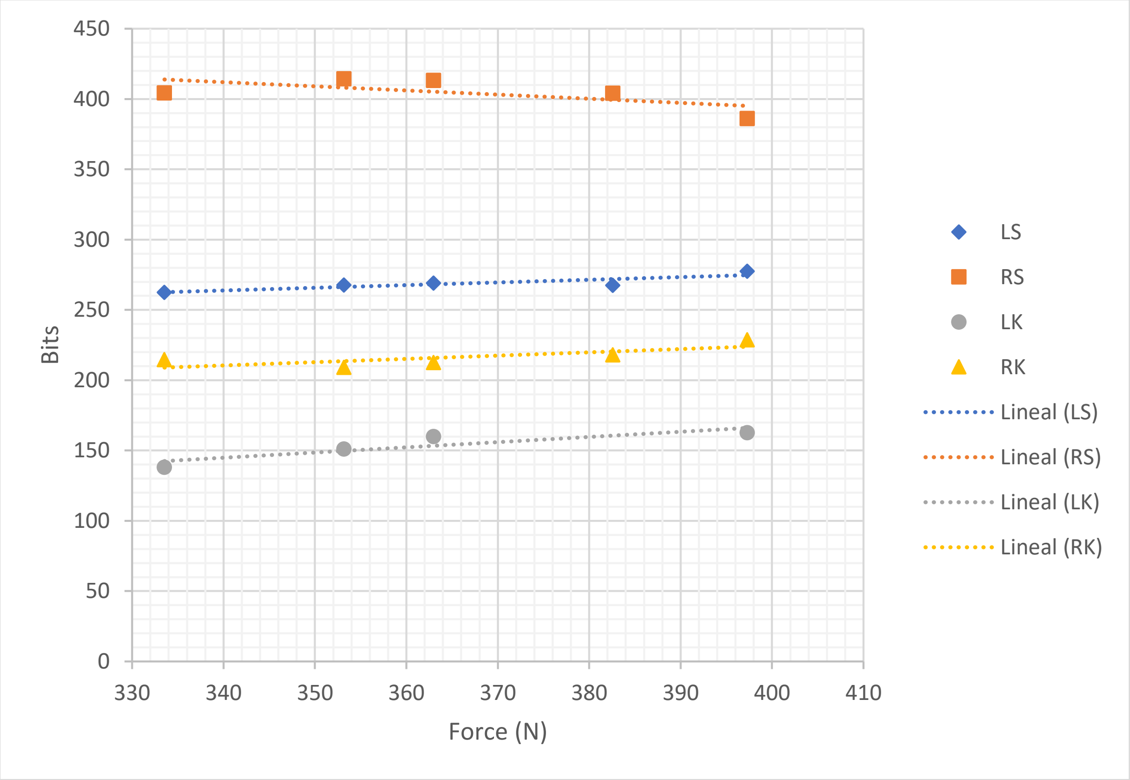


Figure A.2. Test 1. Bits vs. Force: Left Knee.

Appendix B – Constant Force

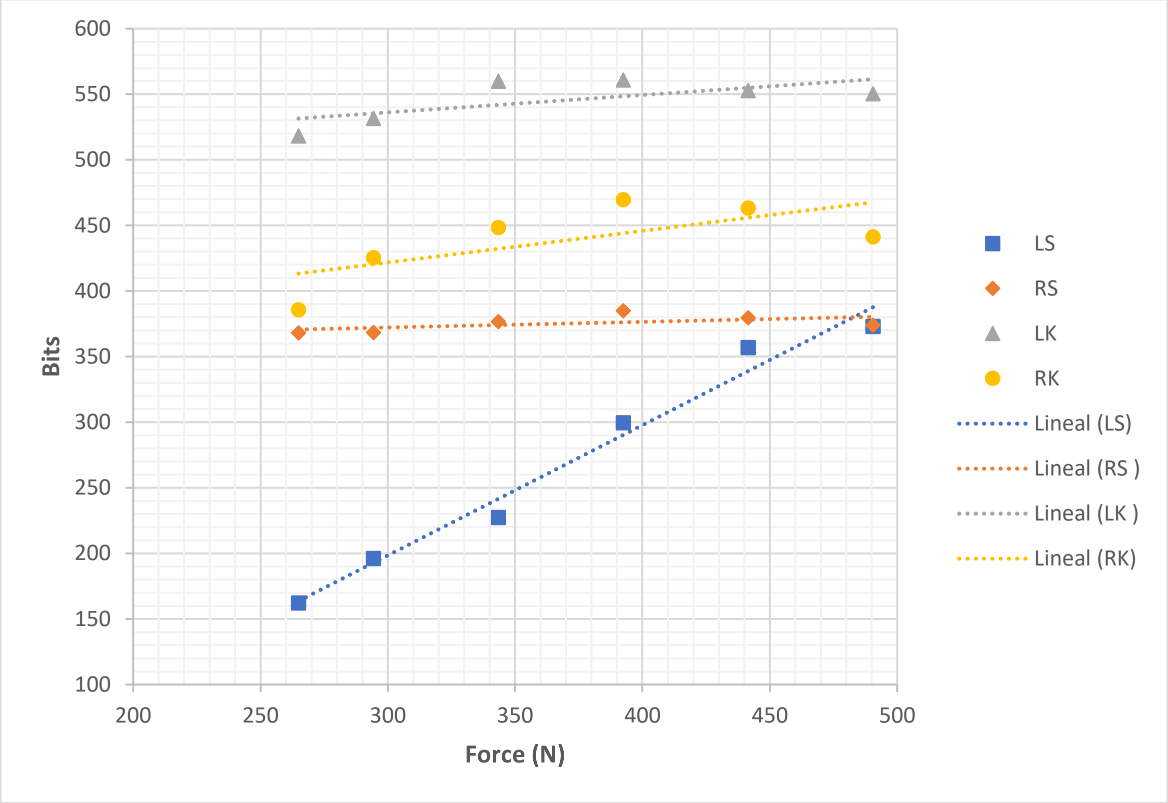


Figure B.1. Test 2. Bits vs. Force: Left Shoulder – trial 2

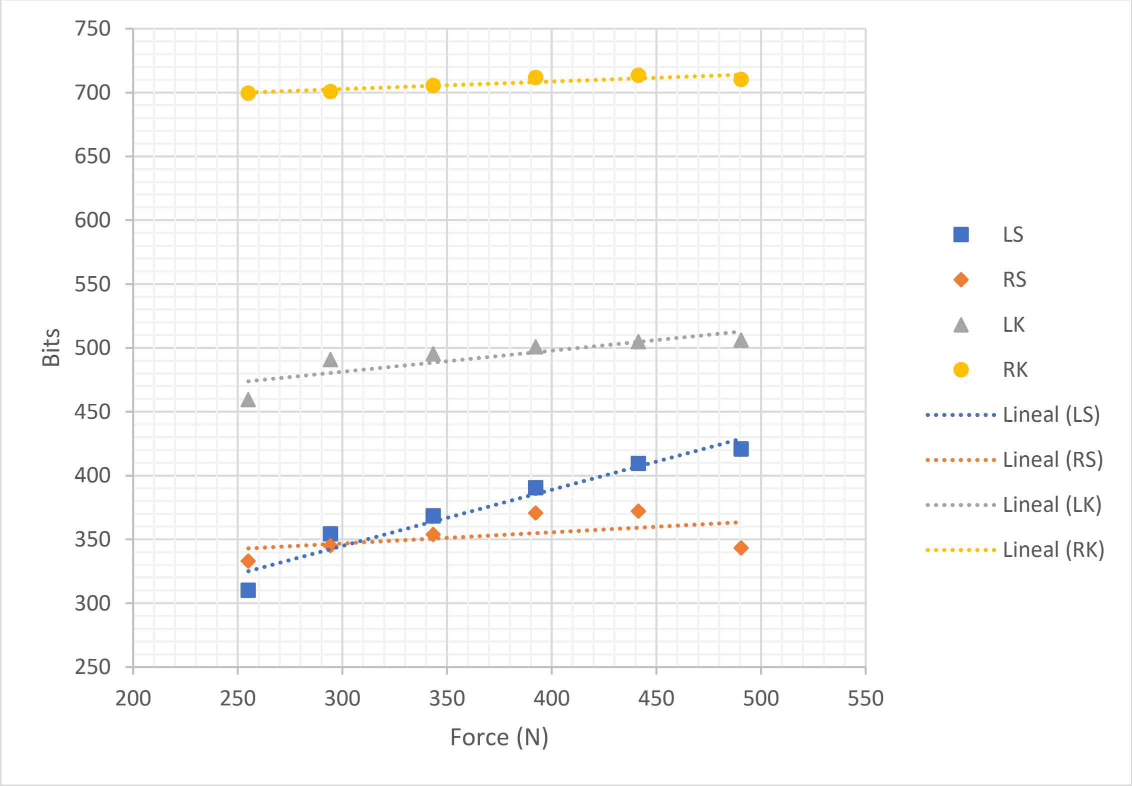


Figure B.2. Test 2. Bits vs. Force: Left Shoulder – trial 3

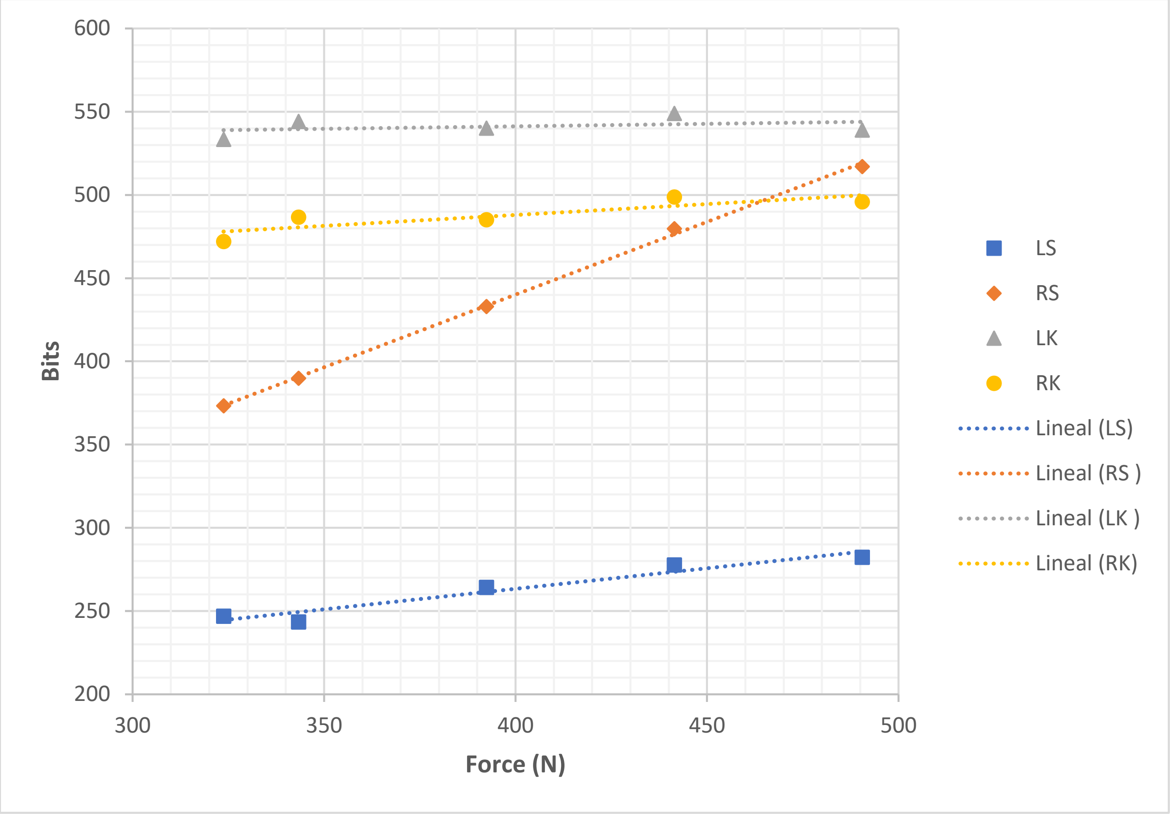


Figure B.3. Test 2. Bits vs. Force: Right Shoulder – trial 2

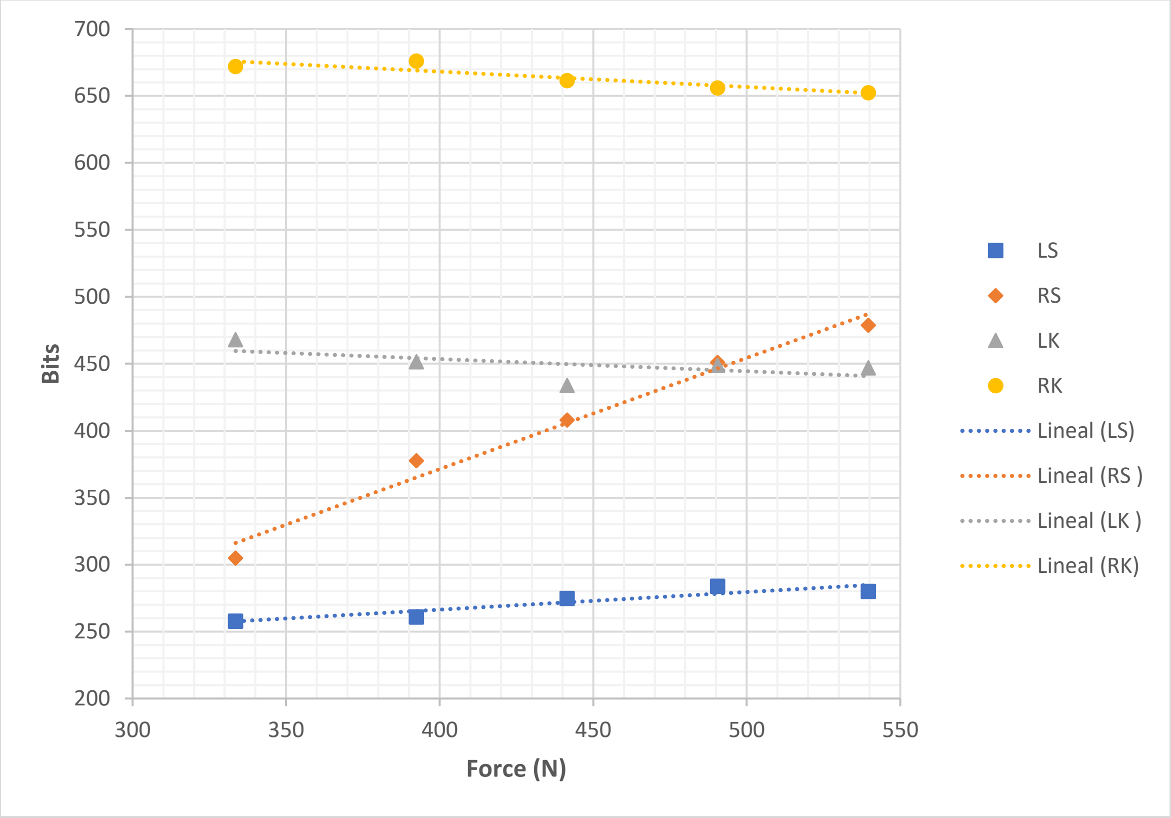


Figure B.4. Test 2. Bits vs. Force: Right Shoulder – trial 3

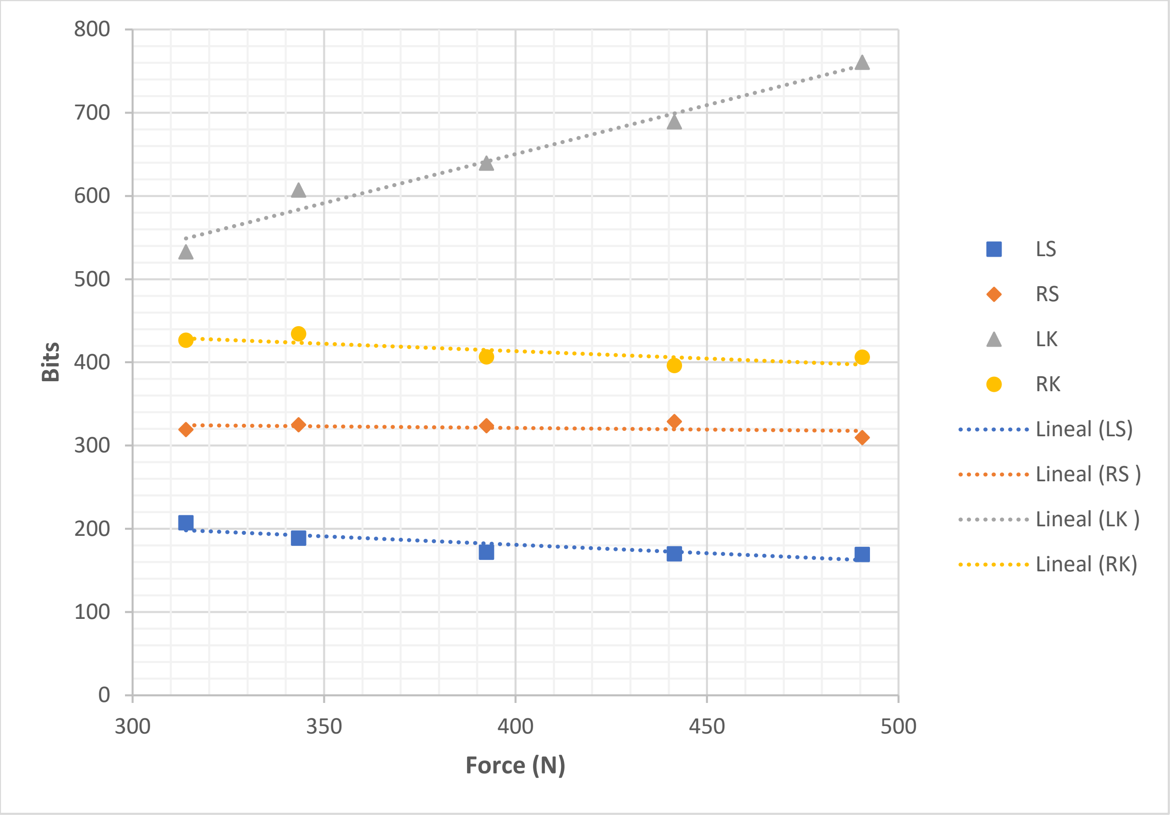


Figure B.5. Test 2. Bits vs. Force: Left Knee – trial 2

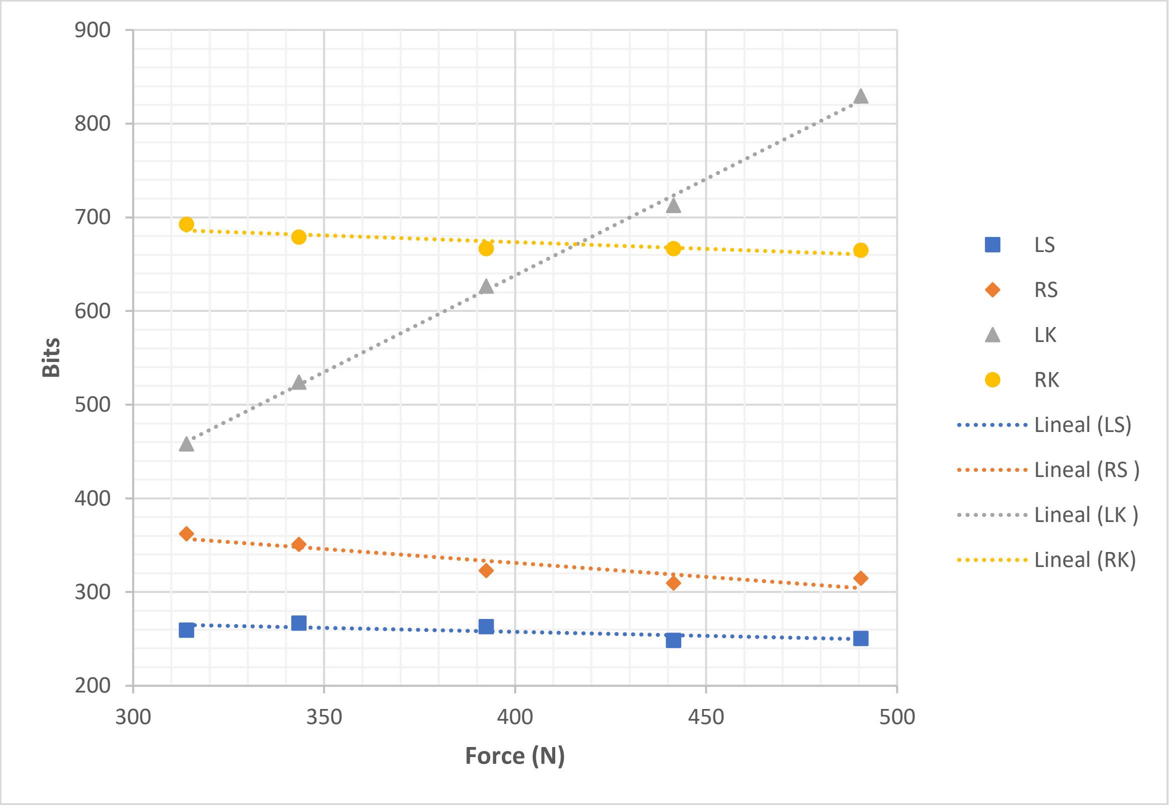


Figure B.6. Test 2. Bits vs. Force: Left Knee – trial 3

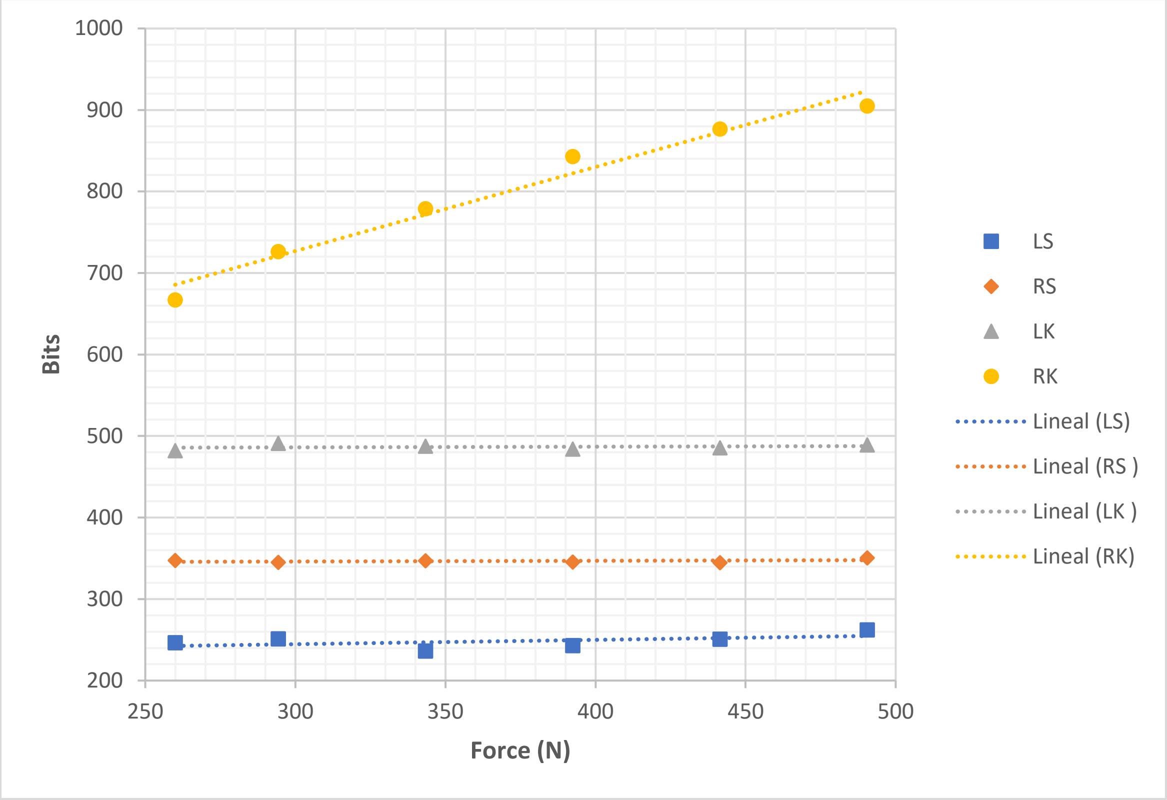


Figure B.7. Test 2. Bits vs. Force: Right Knee – trial 2

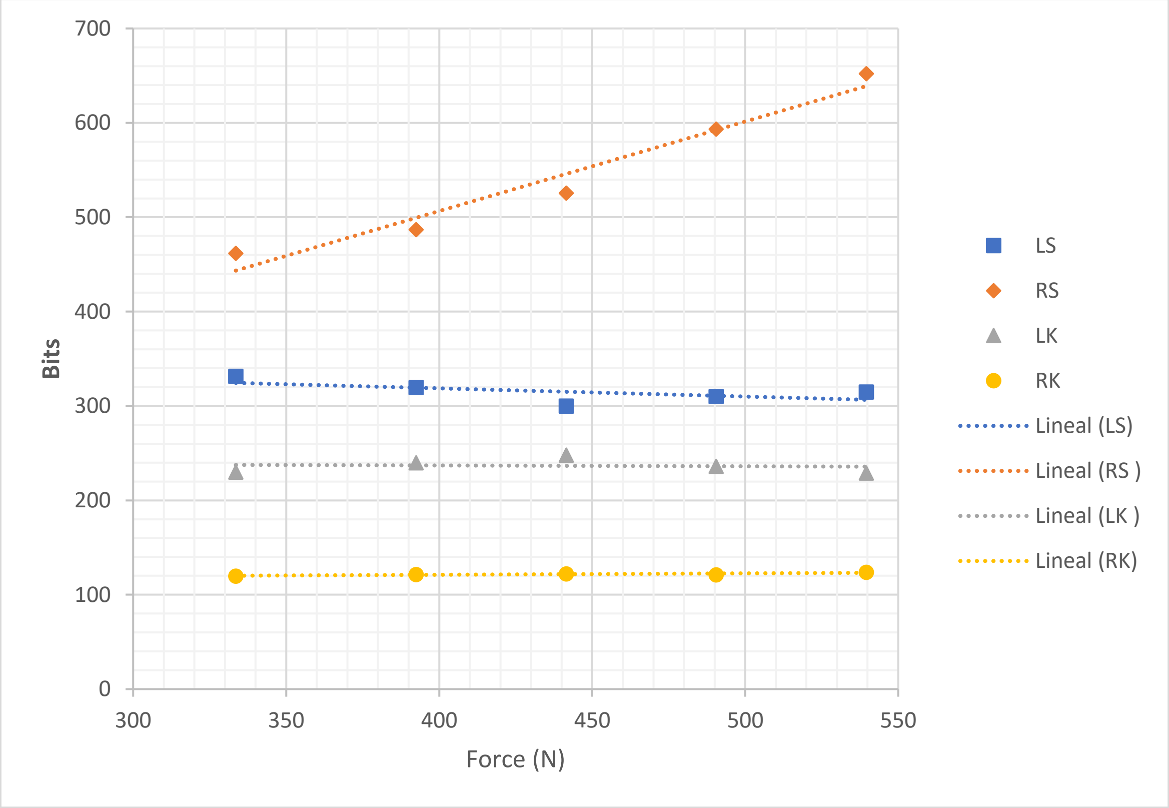


Figure B.8. Test 2. Bits vs. Force: Right Knee – trial 3

Appendix C – Free Steering

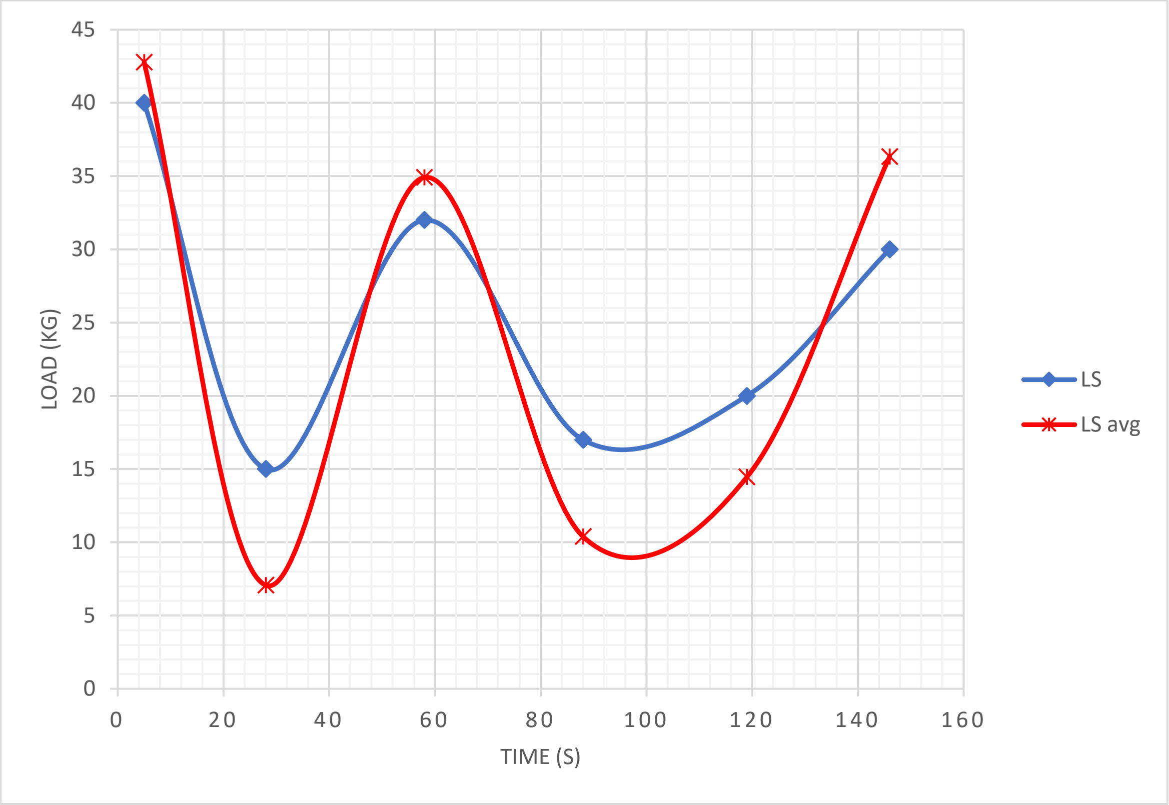


Figure C.1. Free steering test, trial 2. Left shoulder sensor. The blue line (squared markers) represents the load applied and the red line ('x' markers) the calculated load with averaged bits.

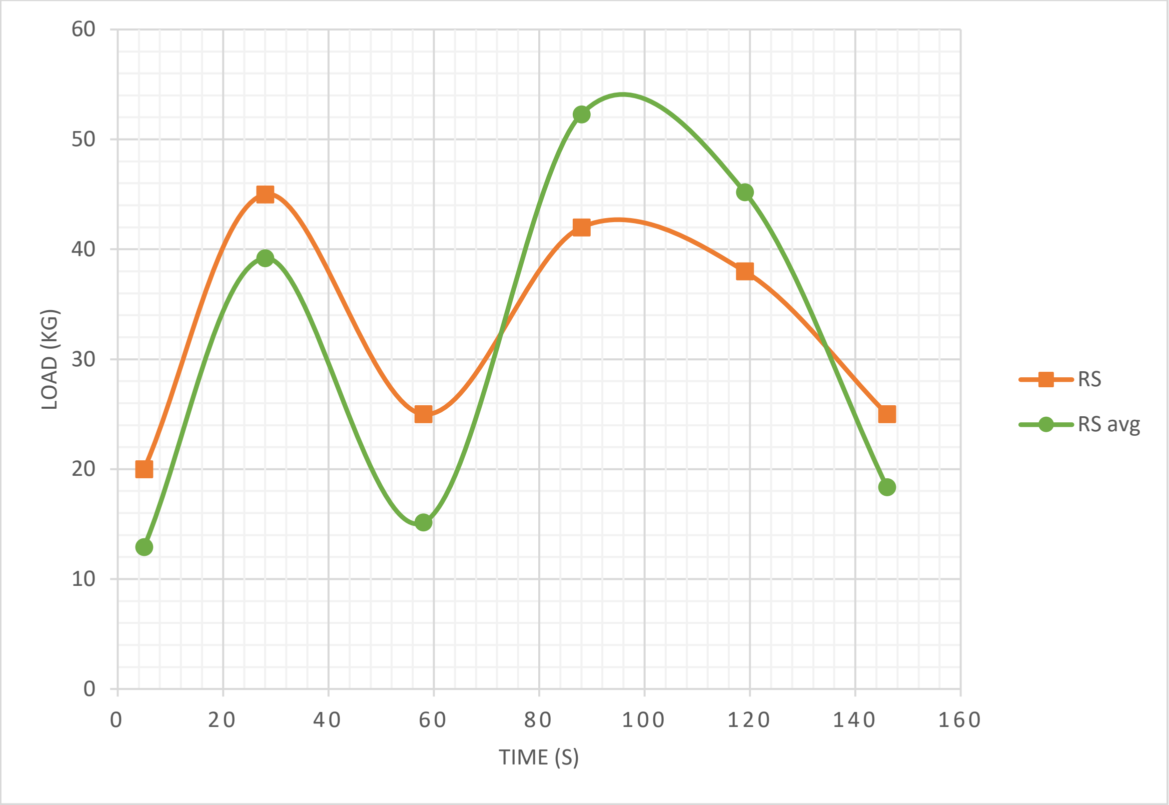


Figure C.2. Free steering test, trial 2. Right shoulder sensors. The orange line (squared markers) represents the load applied and green line (round markers) the calculated load with averaged bits.

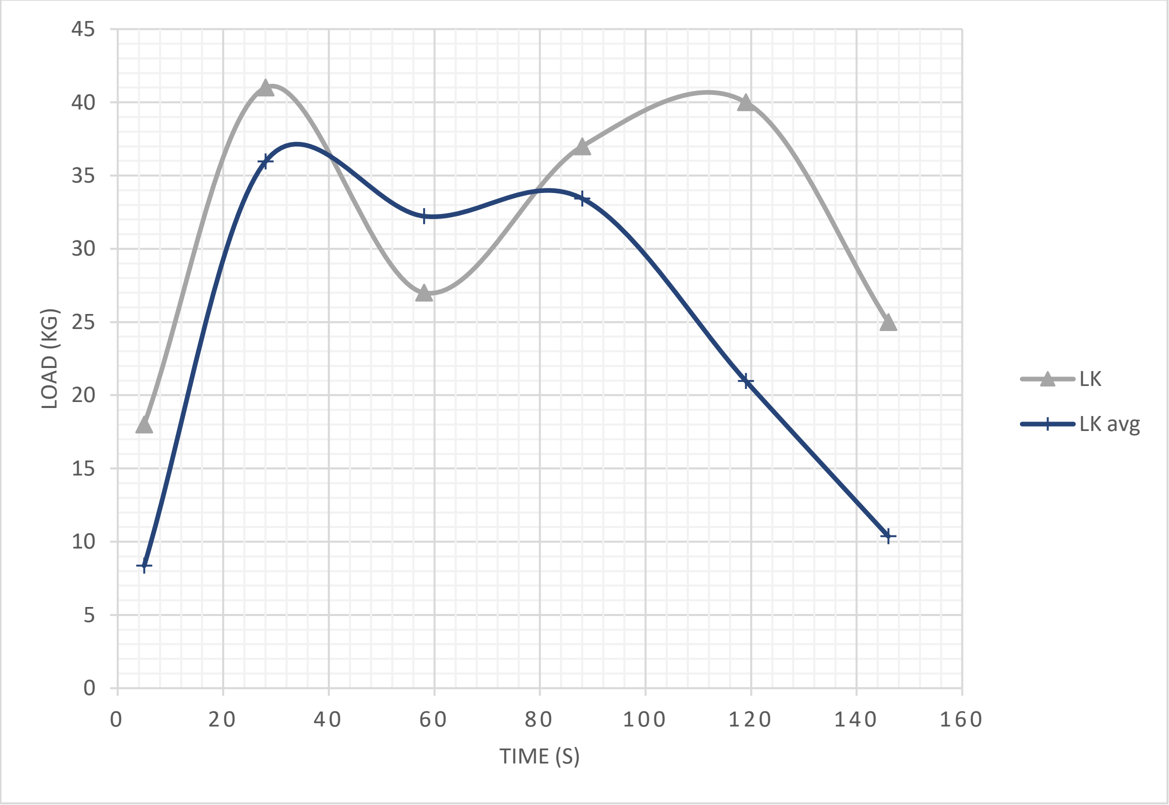


Figure C.1. Free steering test, trial 2. Left knee sensors. Grey line (triangle markers) represents the load applied and the dark blue line (cross markers) the calculated load with averaged bits.

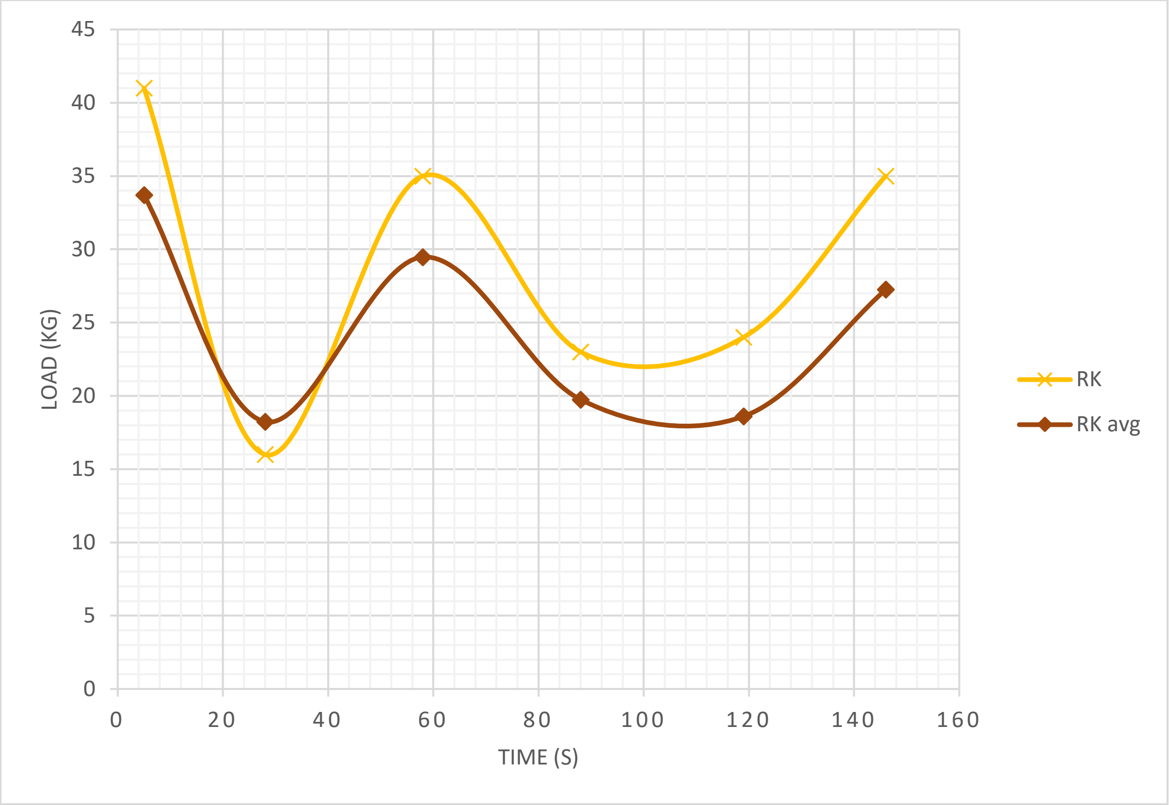


Figure C.4. Free steering test, trial 2. Right knee sensors. Yellow line (round markers) represents the load applied and brown line (diamond line) the calculated load with averaged bits.

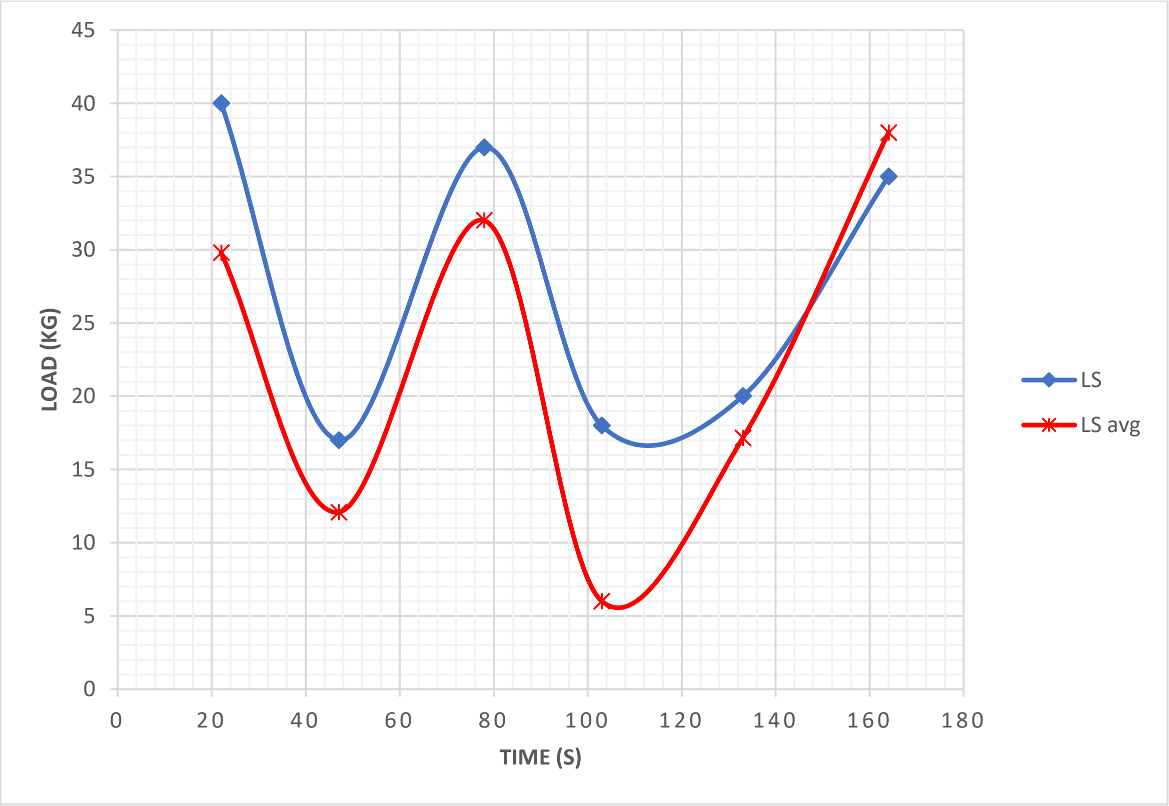


Figure C.5. Free steering test, trial 3. Left shoulder sensor. The blue line (squared markers) represents the load applied and the red line ('x' markers) the calculated load with averaged bits.

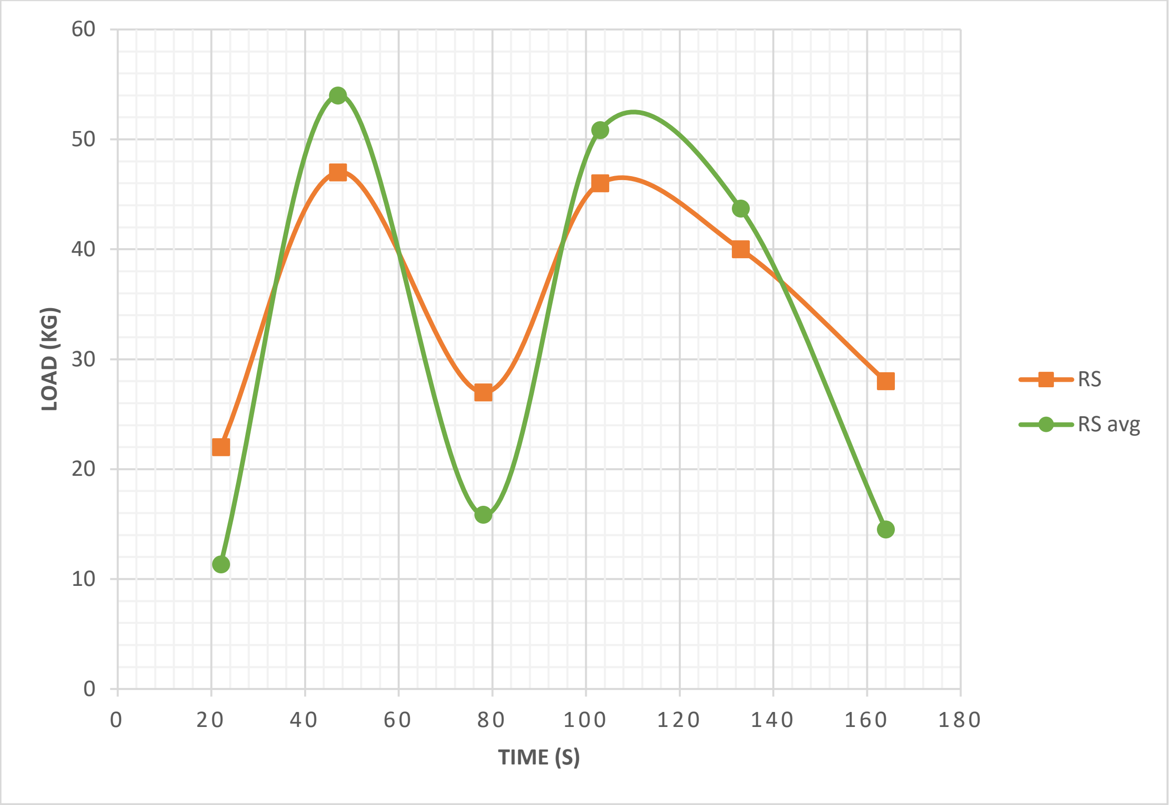


Figure C.6. Free steering test, trial 3. Right shoulder sensors. The orange line (squared markers) represents the load applied and green line (round markers) the calculated load with averaged bits.

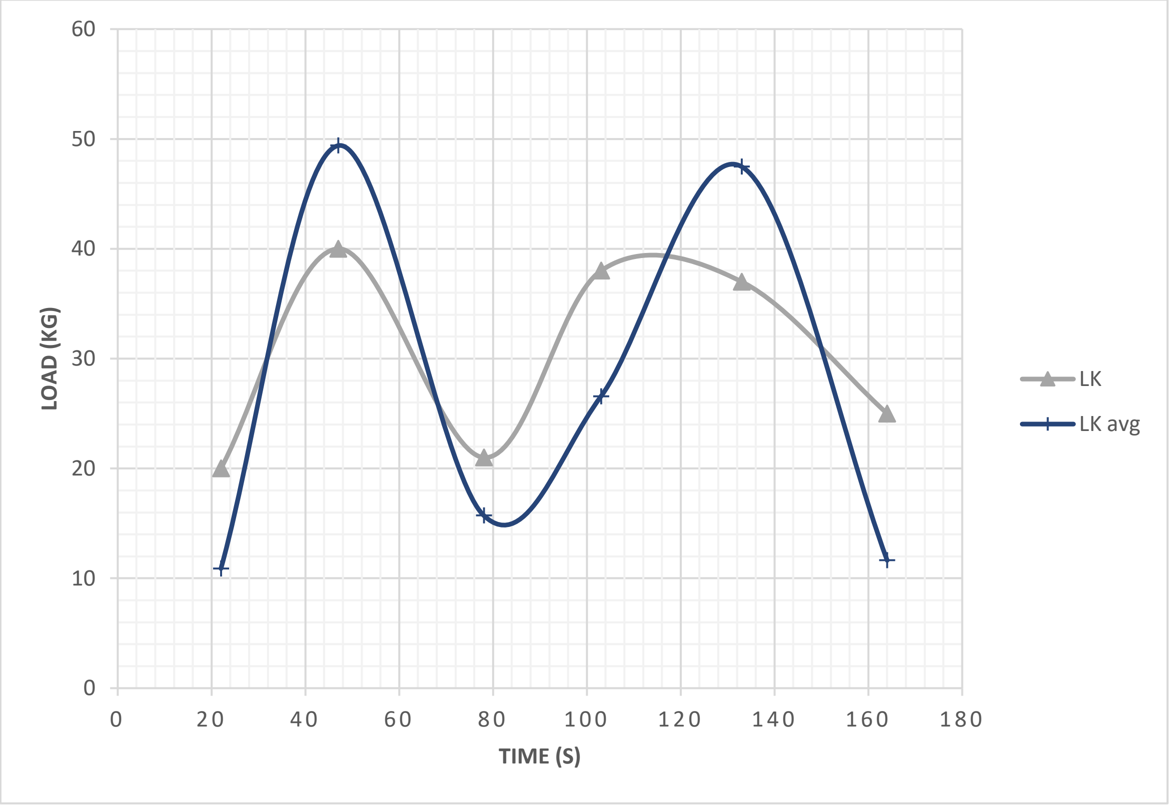


Figure C.7. Free steering test, trial 3. Left knee sensors. Grey line (triangle markers) represents the load applied and the dark blue line (cross markers) the calculated load with averaged bits.

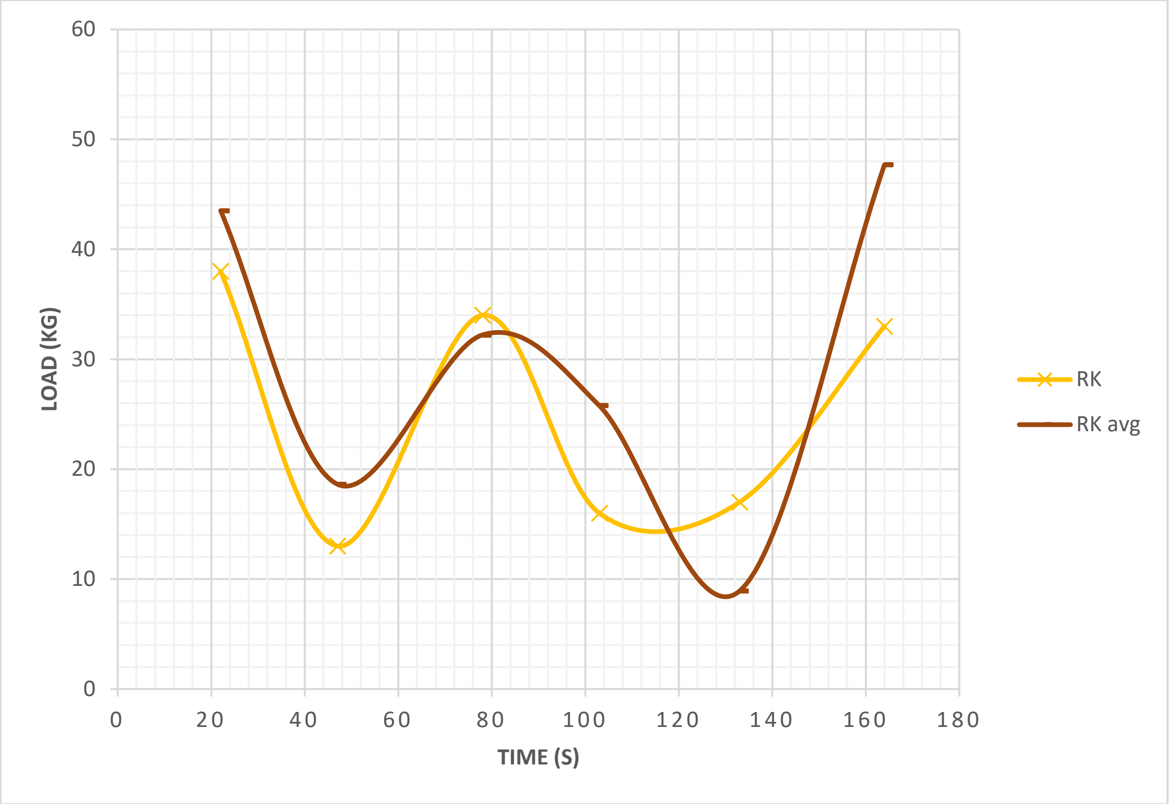


Figure C.8. Free steering test, trial 3. Right knee sensors. Yellow line (round markers) represents the load applied and brown line (diamond line) the calculated load with averaged bits.

Appendix D – Arduino Code

```
// include the SD library:
#include <SPI.h>
#include <SD.h>

// Base name must be six characters or less for short file names.
#define LOGGER "Data"

// change this to match your SD shield or module;
// Arduino Ethernet shield: pin 4
// Adafruit SD shields and modules: pin 10
// Sparkfun SD shield: pin 8
// MKRZero SD: SDCARD_SS_PIN
const int chipSelect = 10;
File dataFile;

const uint8_t BASE_NAME_SIZE = sizeof(LOGGER) - 1;
char fileName[] = LOGGER "00.csv";

//Set up the analog pins used for the sensors
int LS = A0;
int RS = A1;
int LK = A2;
int RK = A3;
unsigned long time;

void setup() {
  // Open serial communications and wait for port to open:
  Serial.begin(9600);

  while (!Serial) {
    ; // wait for serial port to connect. Needed for native USB port only
  }

  Serial.print("Initializing SD card...");

  // see if the card is present and can be initialized:
  if (!SD.begin(chipSelect)) {
    Serial.println("Card failed, or not present");
    // don't do anything more:
    while (1);
  }
  Serial.println("card initialized.");

  // Create a new file everytime the Arduino is powered ON. Files from 00 to 99
  if (!SD.begin(chipSelect)) {
    Serial.println(F("begin failed"));
    return;
  }
  while (SD.exists(fileName)) {
    if (fileName[BASE_NAME_SIZE + 1] != '9') {
      fileName[BASE_NAME_SIZE + 1]++;
    } else if (fileName[BASE_NAME_SIZE] != '9') {
      fileName[BASE_NAME_SIZE + 1] = '0';
      fileName[BASE_NAME_SIZE]++;
    } else {
      Serial.println(F("Can't create file name"));
      return;
    }
  }
  dataFile = SD.open(fileName, FILE_WRITE);
  if (!dataFile) {
    Serial.println(F("open failed"));
    return;
  }
  Serial.print(F("opened: "));
  Serial.println(fileName);
  dataFile.close();


  Serial.println();
  Serial.print("Time(ms)");
  Serial.print("\t");
  Serial.print("LS");
  Serial.print("\t");
  Serial.print("RS");
  Serial.print("\t");
  Serial.print("LK");
  Serial.print("\t");
  Serial.println("RK");
}

void loop() {
  time = millis();
  // Read sensor value
  int LSsensorValue = analogRead(LS);
  int RSsensorValue = analogRead(RS);
  int LKsensorValue = analogRead(LK);
  int RKsensorValue = analogRead(RK);

  //Prints time since the arduino starts reading values every 0.25 s then the read of
the sensor
  //  Serial.print(time);
  //  Serial.print("\t");
  //  Serial.print(LSsensorValue);
  //  Serial.print("\t");
  //  Serial.print("RSsensorValue");
  //  Serial.print("\t");
  //  Serial.print("LKsensorValue");
  //  Serial.print("\t");
  //  Serial.println("RKsensorValue");
  delay (250);

  // make a string for assembling the data to log:
  String dataString = "";

  // open the file. note that only one file can be open at a time,
  // so you have to close this one before opening another.
  dataFile = SD.open(fileName, FILE_WRITE);

  //append to the string + String(AnalogSensorReading)
  dataString = dataString + time;
  dataString = dataString + ", ";
  dataString = dataString + String(LSsensorValue);
  dataString = dataString + ", ";
  dataString = dataString + String(RSsensorValue);
  dataString = dataString + ", ";
  dataString = dataString + String(LKsensorValue);
  dataString = dataString + ", ";
  dataString = dataString + String(RKsensorValue);

  // if the file is available, write to it:
  if (dataFile) {
    dataFile.println(dataString);
    dataFile.close();
    // print to the serial port too:
    Serial.println(dataString);
  }
  // if the file isn't open, pop up an error:
  else {
    Serial.println("error opening datalog.txt");
  }
}
```

Appendix E – GUI Interface

```
classdef SteeringForce_FINAL < matlab.apps.AppBase

    % Properties that correspond to app components
    properties (Access = public)
        UIFigure          matlab.ui.Figure
        UIAxes            matlab.ui.control.UIAxes
        UIAxes4           matlab.ui.control.UIAxes
        UIAxes3           matlab.ui.control.UIAxes
        UIAxes2           matlab.ui.control.UIAxes
        SelectFileButton  matlab.ui.control.StateButton
        EditField         matlab.ui.control.EditField
    end

    % Callbacks that handle component events
    methods (Access = private)

        % Value changed function: SelectFileButton
        function SelectFileButtonValueChanged(app, event)
            value = app.SelectFileButton.Value;

        %Load Data in Bits
        loadBits = uigetfile("*.csv");
        bits = readmatrix(loadBits);
        % Left Shoulder model
        for i = 1:length(bits)
             bits(i,1) = bits(i,1)./1000;
             forceLS(i,:) = (bits(i,2) - 105.66)/0.7205;

            if forceLS(i,:) < 0
               forceLS(i,:)= 0;

            else
            forceLS(i,:)= forceLS(i,:);
            end
        end
        kgLS = forceLS./9.81;
        plot(app.UIAxes, bits(:,1), kgLS, 'LineWidth', 2)

        % Right Shoulder Model
        for j = 1:length(bits)
            forceRS(j,:) = (bits(j,3) - 76.434)/0.8739;

            if forceRS(j,:) < 0
               forceRS(j,:)= 0;

            else
            forceRS(j,:)= forceRS(j,:);
            end
        end
        kgRS = forceRS./9.81;
        plot(app.UIAxes2, bits(:,1), kgRS, 'r', 'LineWidth', 2)

        % Left Knee Model
        for k = 1:length(bits)
```

```
             forceLK(k,:) = (bits(k,4) +65.828)/1.4972;

            if forceLK(k,:) < 0
               forceLK(k,:)= 0;

            else
            forceLK(k,:)= forceLK(k,:);
            end
        end
        kgLK = forceLK./9.81;
        plot(app.UIAxes3, bits(:,1), kgLK, 'g', 'LineWidth', 2)

        % Right Knee Model
        for l = 1:length(bits)

             forceRK(l,:) = (bits(l,5) - 152.41)/1.6237;

            if forceRK(l,:) < 0
               forceRK(l,:)= 0;

            else
            forceRK(l,:)= forceRK(l,:);
            end
        end
        kgRK = forceRK./9.81;
        plot(app.UIAxes4, bits(:,1), kgRK,'m', 'LineWidth', 2)

        [f,n,e] = fileparts(loadBits);
        app.EditField.Value = n;

        end
    end

    % Component initialization
    methods (Access = private)

        % Create UIFigure and components
        function createComponents(app)

            % Create UIFigure and hide until all components are created
            app.UIFigure = uifigure('Visible', 'off');
            app.UIFigure.Position = [100 100 826 562];
            app.UIFigure.Name = 'UI Figure';

            % Create UIAxes
            app.UIAxes = uiaxes(app.UIFigure);
            title(app.UIAxes, 'Left Shoulder')
            xlabel(app.UIAxes, 'Time (s)')
            ylabel(app.UIAxes, 'Force (N)')
            app.UIAxes.Position = [1 320 407 243];

            % Create UIAxes4
            app.UIAxes4 = uiaxes(app.UIFigure);
            title(app.UIAxes4, 'Right Knee')
            xlabel(app.UIAxes4, 'Time (s)')
            ylabel(app.UIAxes4, 'Force (N)')
            app.UIAxes4.Position = [407 69 412 252];

            % Create UIAxes3
            app.UIAxes3 = uiaxes(app.UIFigure);
            title(app.UIAxes3, 'Left Knee')
            xlabel(app.UIAxes3, 'Time (s)')
            ylabel(app.UIAxes3, 'Force (N)')
            app.UIAxes3.Position = [1 69 407 252];

            % Create UIAxes2
            app.UIAxes2 = uiaxes(app.UIFigure);
            title(app.UIAxes2, 'Right Shoulder')
            xlabel(app.UIAxes2, 'Time (s)')
            ylabel(app.UIAxes2, 'Force (N)')
            app.UIAxes2.Position = [407 320 418 243];

            % Create SelectFileButton
            app.SelectFileButton = uibutton(app.UIFigure, 'state');
            app.SelectFileButton.ValueChangedFcn = createCallbackFcn(app, @SelectFileButtonValueChanged,
true);
            app.SelectFileButton.Text = 'Select File';
            app.SelectFileButton.Position = [268 32 100 19];

            % Create EditField
            app.EditField = uieditfield(app.UIFigure, 'text');
            app.EditField.Position = [397 31 240 20];

            % Show the figure after all components are created
            app.UIFigure.Visible = 'on';
        end
    end

    % App creation and deletion
    methods (Access = public)

        % Construct app
        function app = SteeringForce_FINAL

            % Create UIFigure and components
            createComponents(app)

            % Register the app with App Designer
            registerApp(app, app.UIFigure)

            if nargout == 0
                clear app
            end
        end

        % Code that executes before app deletion
        function delete(app)

            % Delete UIFigure when app is deleted
            delete(app.UIFigure)
        end
    end
end
```

References

[1] I. J. M. Roberts, “Skeleton bobsleigh mechanics: athlete-sled interaction,” Jul. 2013.

[2] FIBT, “F.I.B.T International Skeleton Rules,” 2007. [Online]. Available: www.bobsleigh.com. [Accessed: 08-Oct-2018].

[3] E. Sabbioni, S. Melzi, F. Cheli, and F. Braghin, “Bobsleigh and Skeleton,” in *The Engineering Approach to Winter Sports*, New York, NY: Springer New York, 2016, pp. 183–276.

[4] E. P. Lozowski, K. Szilder, S. Maw, and A. Morris, “A model of ice friction for skeleton sled runners,” *Proc. Int. Offshore Polar Eng. Conf.*, 2014.

[5] BromleySports, “Bromley Sled.” [Online]. Available: http://bromleysports.com/sports/skeleton/p14. [Accessed: 22-Aug-2019].

[6] IBSF, “International Skeleton Rules 2018,” 2018. [Online]. Available: https://www.ibsf.org/images/documents/downloads/Rules/2018_2019/2018_International_Rules_SKELETON-am.pdf. [Accessed: 21-Aug-2019].

[7] C. Chi, X. Sun, N. Xue, T. Li, and C. Liu, “Recent Progress in Technologies for Tactile Sensors.,” *Sensors (Basel).*, vol. 18, no. 4, Mar. 2018.

[8] K. Weiss and H. Worn, “The working principle of resistive tactile sensor cells,” in *IEEE International Conference Mechatronics and Automation, 2005*, pp. 471–476.

[9] F. Sygulla, F. Ellensohn, A.-C. Hildebrandt, D. Wahrmann, and D. Rixen, “A flexible and low-cost tactile sensor for robotic applications,” in *2017 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, 2017, pp. 58–63.

[10] Y. Wan, Y. Wang, and C. F. Guo, “Recent progress on flexible tactile sensors,” *Mater. Today Phys.*, vol. 1, pp. 61–73, Jun. 2017.

[11] A. Young, “Arduino Force Sensing Resistor (FSR)No Title,” 2019. [Online]. Available: https://pimylifeup.com/arduino-force-sensing-resistor/. [Accessed: 14-May-2019].

[12] User, “Shimmer User Manual Revision 3p,” 2017.

[13] “Wearable IMU sensor | Motion sensor | 9DOF.” [Online]. Available: https://www.shimmersensing.com/products/shimmer3-imu-sensor. [Accessed: 12-Nov-2019].